Rebuild and Ensemble: Exploring Defense Against Text Adversaries

Anonymous ACL submission

Abstract

Adversarial attacks can mislead strong neural models; as such, in NLP tasks, substitution-based attacks are difficult to defend. Current defense methods usually assume that the substitution candidates are accessible, which cannot be widely applied against substitution-agnostic attacks. In this paper, we propose a **Rebuild and Ensemble** Framework to defend against adversarial attacks in texts without knowing the candidates. We propose a rebuild mechanism to train a robust model and ensemble the rebuilt texts during inference to achieve good adversarial defense results. Experiments show that our method can improve accuracy under the current strong attack methods.

1 Introduction

Adversarial examples (Goodfellow et al., 2014) can successfully mislead strong neural models in both computer vision tasks (Carlini and Wagner, 2016) and language understanding tasks (Alzantot et al., 2018; Jin et al., 2019). An adversarial example is a maliciously crafted example attached with an imperceptible perturbation and can mislead neural networks. To defend attack examples of images, the most effective method is adversarial training (Goodfellow et al., 2014; Madry et al., 2019) which is a mini-max game used to incorporate perturbations into the training process.

Defending adversarial attacks is extremely important in improving model robustness. However, defending adversarial examples in natural languages is more challenging due to the discrete nature of texts. That is, gradients cannot be used directly in crafting perturbations. The generation process of substitution-based adversarial examples is more complicated than using gradient-based methods in attacking images, making it difficult for neural networks to defend against these substitution-based attacks:

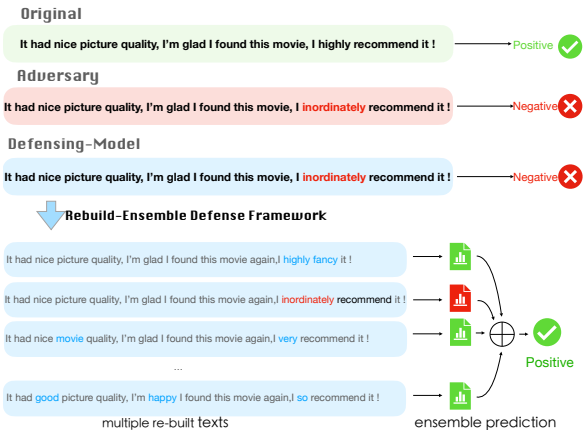


Figure 1: Illustration of Adversarial Defense

(A) The first challenge of defending against adversarial attacks in NLP is that due to the discrete nature, these substitution-based adversarial examples can have substitutes in any token of the sentence and each substitute has a large candidate list. This would cause a combinatorial explosion problem, making it hard to apply adversarial training methods. Strong attacking methods such as Jin et al. (2019) show that using the crafted adversarial examples as data augmentation in adversarial training cannot effectively defend against these substitution-based attacks.

(B) Further, the defending strategies such as adversarial training rely on the assumption that the candidate lists of the substitutions are accessible. However, the candidate lists of the substitutions should **not** be exposed to the target model; that is, the target model should be unfamiliar to the *candidate-agnostic* adversarial examples. In real-world defense systems, the defender is not aware of the strategy the potential attacks might use, so the assumption that the candidate list is available would significantly constrain the potential applications of these defending methods.

In this work, we propose a strong defense framework, i.e., **Rebuild and Ensemble**.

We aim to construct a defense system that can successfully defend the attacks launched by strong methods such as Textfooler (Jin et al., 2019) and BERT-Attack (Li et al., 2020) without expecting of the incoming of these attacks. We introduce a rebuild and ensemble process, we assume that we can reconstruct a clean input sample that does not the adversarial effect based on possible adversarial input. As seen in Figure 1, when the input is changed by the adversarial attack, we can first rebuild the input texts and then make predictions based on the rebuilt texts which will results in correct predictions.

To achieve this goal, we first reconsider the widely applied pre-trained models exemplified by BERT (Devlin et al., 2018) which introduces the masked language modeling task in the pre-training stage and can be used in fine-tuning on downstream tasks. During downstream task fine-tuning, these pre-train models throw away the the learned language modeling ability and focus on making downstream task predictions. Instead of simply fine-tuning downstream tasks, we keep the mask prediction ability during fine-tuning, and use this ability to process the rebuilding of input texts. That is, we random mask the input texts and use the mask prediction to rebuild a text that does not have adversarial affect. Intuitively, the rebuild process introduces randomness since the masks are randomly selected, we can make multiple random rebuilt texts and apply an ensemble process to obatin the final model output predictions for better robustness. To train the defending framework, we introduce the rebuild training based on virtual input adversarial training methods to enhance both rebuilding and downstream task predicting abilities.

Through extensive experiments, we prove that the proposed defense framework can successfully resist strong attacks such as Textfooler and BERT-Attack. Experiments results show that the accuracy under attack in baseline defense methods is lower than random guesses, while ours can lift the performances to only a few percent lower than the original accuracy when the candidates are limited. Further, extensive results indicate that the candidate size of the attacker score is essential for successful attacks, which is a key factor in maintaining semantics of the adversaries. Therefore we also recommend that future attacking methods can focus on achieving success attacks with tighter constrains.

To summarize our contributions:

- We raise the concerns of defending *candidate-agnostic* attacks in NLP tasks.
- We propose a Rebuild and Ensemble framework to defend against recently introduced strong attack methods without knowing the candidates and experiments prove the effectiveness of the framework.
- We explore the key factors in defending against score-based attacks and recommend further research to focus on tighter constraint attacks.

2 Related Work

2.1 Adversarial Attacks in NLP

In NLP tasks, current methods use substitution-based strategies (Alzantot et al., 2018; Jin et al., 2019; Ren et al., 2019) to craft adversarial examples. Most works focus on the score-based black-box attack, that is, the attacking method knows the logits of the output prediction. These methods use different strategies (Yoo et al., 2020; Morris et al., 2020b) to find words to replace such as generic algorithm (Alzantot et al., 2018), greedy-search (Jin et al., 2019; Li et al., 2020) or gradient-based (Ebrahimi et al., 2017; Cheng et al., 2019) and get substitutes using synonyms (Jin et al., 2019; Mrkšić et al., 2016; Ren et al., 2019) or language models (Li et al., 2020; Garg and Ramakrishnan, 2020; Shi et al., 2019).

2.2 Adversarial Defenses

There are fewer methods focusing on defending against adversarial attacks in NLP compared with various types of adversarial attacks.

Under the candidate-agnostic attacker setting, Samangouei et al. (2018) uses a defensive GAN framework to build clean images to avoid adversarial attacks; Xie et al. (2017) introduces randomness into the model predicting process to mitigate adversarial affect. Ebrahimi et al. (2017); Cheng et al. (2019) introduces gradient-based adversarial training that craft adversarial samples by finding the most similar word embeddings based on the gradients. Further, gradient-based virtual adversarial training could also be used in the NLP tasks: Miyato et al. (2016) proposes a virtual adversarial training process, which is later explored in model robustness (Zhu et al., 2019; Li and Qiu, 2020). Basically, they incorporate gradients to craft virtual adversaries to apply robust training.

To defend against adversaries under the candidate-aware assumption, augmentation-based methods are the most direct defense strategies that use the generated adversaries to train a robsut model (Jin et al., 2019; Li et al., 2020; Si et al., 2020). Jia et al. (2019); Huang et al. (2019) introduces a certified robust model to defend against adversarial attacks by constructing a certified space that can tolerate substitutes. Zhou et al. (2020); Dong et al. (2021) construct a convex hull based on the candidate list of that can resist substitutions in the candidate list. Zhou et al. (2019) incorporate the idea of blocking adversarial attacks by discriminating perturbations in the input texts.

3 Rebuild And Ensemble as Defense

Defending against adversarial attacks without accessing the candidate list is more applicable in real-world adversarial defenses. Therefore, we introduce **Rebuild and Ensemble** as an effective framework to defend strong adversarial attacks exemplified by substitution-based attacks in NLP without knowing the candidate list of substitutions.

Without knowing the candidates, the model cannot resist the substitutes that have strong adversarial affect. Therefore, the model needs to avoid the adversaries by replacing them with clean texts. So we introduce the rebuild process therefore when the target model is facing adversaries, it can first reconstruct the inputs to avoid facing the spears of the adversarial examples.

We suppose that the target model is a fine-tuned model that has been pre-trained by mask language models that may face the adversarial attack is a classification model $F_c(\cdot)$. When given an input sentence X, the adversarial attack may craft an adversarial example X_{adv} that replaces a small proportion of tokens with similar texts. We only consider substitution-based adversaries since the strategy of defending other types of adversarial examples such as token insertion or deletion is exactly the same as defending substitution-based adversaries.

3.1 Rebuild and Ensemble Framework

We propose the rebuild and ensemble framework that first makes multiple input texts and then use these rebuilt texts to make predictions.

We have a model that can re-build input texts and make predictions so we use $F_m(\cdot)$ to denote the mask prediction task that rebuild the input texts and use $F_c(\cdot)$ to denote the classification task.

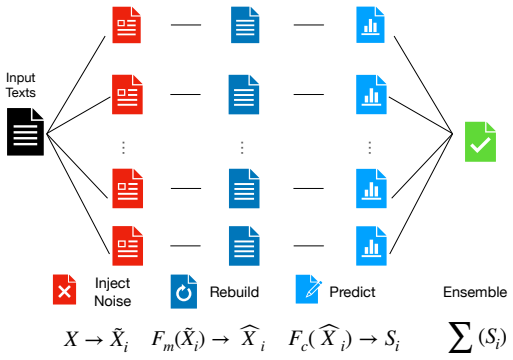


Figure 2: Rebuild And Ensemble Process

As seen in Figure 2, when given an input text $X = [w_0, \cdots, w_n,]$ that might have been attacked, we random mask the input texts or insert additional masks to make N copies of noisy input $\tilde{X}_i = [w_0, \cdots, \texttt{[MASK]}, w_n, \cdots,]$. We use two simple strategies to inject noise into the input texts: (1) Randomly mask the input texts; (2) Randomly insert masks into the input texts.

After making multiple noisy inputs, we can run the rebuild process first to get the rebuilt texts based on the randomly masked inputs $\tilde{X}$:

$$\widehat{X}_i = F_m(\tilde{X}_i) \tag{1}$$

Then we feed the rebuilt texts through the classifier $F_c(\cdot)$ to calculate the final output score based on the multiple rebuilt texts:

$$S_i = \frac{1}{N}\sum_{i=0}^{N}\Big(Softmax(F_c(\widehat{X}_i))\Big) \tag{2}$$

Here, we use the average score from multiple rebuilt texts predictions as the final output score given to the score-based adversarial attackers.

Besides, in the rebuild process, we aim to make best use of the mask prediction ability that the pre-trained models possess since the fine-tuning process only uses limited downstream task data while the pre-training stage includes massive data and calculation which can be helpful in better robustness against text adversaries.

3.2 Rebuild Framework Training

We use the fine-tuned masked language model while maintaining the mask language modeling ability since we believe that (1) rebuild process can help gain better robustness by mitigating the adversarial affect in the input sequences; (2) maintaining language modeling information helps improve model robustness in the classification process.

In order to fine-tune such a model F with parameter θ containing two functions $F_m(\cdot)$ and $F_c(\cdot)$, we introduce a rebuild training process based on multi-task adversarial training. We use noisy texts as inputs to train the mask language modeling task and the downstream task fine-tuning simultaneously so that the fine-tuning process can tolerate more noisy texts since the model might be attacked by adversaries.

3.2.1 Mask LM Training Strategy

In our model fine-tuning, we have both the mask language modeling training and the downstream task training. In the mask language model training, we also incorporate the gradient information in the rebuild training process to build a gradient-based noisy data to enhance the rebuilding ability.

Therefore we have two language model training strategy:

(1) Standard `[MASK]` Prediction: We randomly mask the input texts and make the masked language model to further pre-train the masks on the training dataset.

(2) Gradient-Noise Rebuild: Previous pre-training process does not calculate loss on un-masked tokens. Instead we use a gradient-based adversarial training method to add perturbation δ on the embedding space of these un-masked tokens and calculate the loss of the masked language model task on these tokens to make the model aware of the potential substitutes.

3.2.2 Preliminary of Gradient-Based Adversarial Training

Recent researches have been focusing on exploring the possibility of using gradient-based virtual adversaries in NLP tasks (Zhu et al., 2019; Li and Qiu, 2020). The core idea is that the adversarial examples are not real substitutions but virtual perturbations:

$$\boldsymbol{\delta} = \prod_{||\boldsymbol{\delta}||_F \leq \epsilon} \frac{\alpha g(\boldsymbol{\delta})}{||g(\boldsymbol{\delta})||_F} \tag{3}$$

$$g(\boldsymbol{\delta}) = \bigtriangledown_{\boldsymbol{\delta}} L(f_\theta(\boldsymbol{X} + \boldsymbol{\delta}), y) \tag{4}$$

Here $\prod_{||\boldsymbol{\delta}||_F \leq \epsilon}$ represents the process that projects the perturbation onto the normalization ball ϵ using Frobenius normalization $||\boldsymbol{\delta}||_F$. We update the perturbation using a certain adversarial learning rate α. X is the word embedding of input sequence $[w_0, \cdots, w_n,]$. The perturbation is not actual substitutions, but adversarial embeddings. Then these virtual adversaries are used in the training process to improve model performances.

Algorithm 1 Rebuild Training

Require: Training Sample X

1: $\boldsymbol{\delta} \leftarrow \frac{1}{\sqrt{D}} U(-\sigma, \sigma)$ // Initialize Perturbation
2: $\tilde{X} \leftarrow$ Random Mask X
3: $\mathcal{L}_c \leftarrow$ Using Equation 5
4: $\mathcal{L}_{mlm} \leftarrow$ Using Equation 6
5: // Get Perturbation
6: $\boldsymbol{g}_\delta \leftarrow \bigtriangledown_\delta (\mathcal{L}_c + \mathcal{L}_{mlm})$
7: $\boldsymbol{\delta} \leftarrow \prod_{||\boldsymbol{\delta}||_F < \epsilon} (\boldsymbol{\delta} + \alpha \cdot \boldsymbol{g}_\delta / ||\boldsymbol{g}_\delta||_F)$
8: // Rebuild with Noise
9: $\mathcal{L}_{noise} \leftarrow$ Using Equation 8
10: $\boldsymbol{g} = \bigtriangledown_\theta (\mathcal{L}_c + \mathcal{L}_{mlm} + \mathcal{L}_{noise})$
11: $\theta \leftarrow \theta - \boldsymbol{g}$ // Update model parameter θ

3.2.3 Overall Process of Rebuild Training

Given input texts X, we first make noisy copies $\tilde{X}$, for notation convenience, here X and $\tilde{X}$ are the embedding output of the input texts. Then we can calculate the gradients of the fine-tuning classification task $\boldsymbol{g}_c$ as well as the mask-prediction task $\boldsymbol{g}_{mlm}$.

$$\mathcal{L}_c = L(F_c(\tilde{X}), y, \theta) + L(F_c(X), y, \theta) \tag{5}$$

$$\mathcal{L}_{mlm} = L(F_m(\tilde{X}), X, \theta) \tag{6}$$

Here, L is the cross entropy loss function for both masked language model task $\mathcal{L}_{mlm}$ and classification task $\mathcal{L}_c$. As seen in Algorithm 1 line 6, we run the fine-tuning process based on the noisy input and the original input and we run the mask prediction task simultaneously. We assume that with the mask prediction task also involved in fine-tuning, the model will not be focusing on fitting the classification task only, which can help maintain the entire semantic information and mitigate the adversarial affect from the adversaries.

Further, we use gradients to craft virtual adversaries δ and calculate loss based on these adversaries $\mathcal{L}_{noise}$:

$$\boldsymbol{\delta} \leftarrow \prod_{||\boldsymbol{\delta}||_F < \epsilon} (\boldsymbol{\delta} + \alpha \cdot \boldsymbol{g}_\delta / ||\boldsymbol{g}_\delta||_F) \tag{7}$$

$$\mathcal{L}_{noise} = L(F_m(\tilde{X} + \boldsymbol{\delta}), X, \theta) \tag{8}$$

Here the cross entropy loss L is calculated based on all tokens not just the masked ones. In this way, the masked language model prediction task is modified to make the model tolerate more noises and therefore more robust.

The difference between our rebuild-training and traditional virtual adversarial training is that we allow the perturbations to be extremely **large**. That is, the adversarial learning rate α and the perturbation boundary ϵ are larger than those used in the FreeLB and TAVAT method. Therefore, some of the tokens are seriously affected by gradients, which is an effective method for further pre-training the model to tolerate adversaries. Further, the perturbations δ are based on both prediction loss and the language model loss, which cover a wider range so that the model can be more resilient.

Given training batch B, we calculate all the losses of prediction task, rebuild task and gradient-based noise rebuild task and update the model parameter. Therefore, we can train a model that can rebuild the input texts from a noisy input, also it can make robust predictions based on the rebuilt texts.

With the proposed rebuild training, we can improve the model robustness in two perspectives: (1) we have a more robust forward process since the model will first rebuild the potentially sabotaged texts and predict the model label based on the rebuilt texts; (2) we have a more robust model since the fine-tuned model possesses more semantic information than normal fine-tuned models;

4 Experiments

4.1 Datasets

We use two widely used text classification tasks: IMDB [1] (Maas et al., 2011) and AG's News [2] (Zhang et al., 2015) in our experiments. The IMDB dataset is a bi-polar movie review classification task; the AG's News dataset is a four-class news genre classification task. The average length is 220 words in the IMDB dataset, and 40 words in the AG's News dataset. We use the test set following the Textfooler 1k test set in the main result and sample 100 samples for the rest of the experiments since the attacking process is seriously slowed down when the model is defensive.

[1]https://datasets.imdbws.com/

[2]https://www.kaggle.com/amananandrai/ag-news-classification-dataset

4.2 Attack Methods

Popular attack methods exemplified by Generic Algorithm (Alzantot et al., 2018), Textfooler (Jin et al., 2019) and BERT-Attack (Li et al., 2020) can successfully mislead strong models of both IMDB and AG's News task with a very small percentage of substitutions. Therefore, we use these strong adversarial attack methods as the attacker to test the effectiveness of our defense method. The hyper parameters used in the attacking algorithm vary in different settings: we choose candidate list size K to be 12 and 48 typically which are used in the Textfooler and BERT-Attack methods.

We use the IMDB task and the AG's News task since the average sequence length is relatively long. These long-texts tasks are more vulnerable under adversarial attacks since the perturbation rate is considerably small. Normally, we assume that a small percent of substitutes should not drastically change the classification results.

4.3 Victim Models and Defense Baselines

The victim model is the fine-tuned pre-train models exemplified by BERT and RoBERTa, which we implement based on Huggingface Transformers [3] (Wolf et al., 2020). As discussed above, there are few works concerning adversarial defenses against candidate-agnostic attacks in NLP tasks. Moreover, previous works do not focus on recent strong attack algorithms such as Textfooler (Jin et al., 2019), BERT-involved attacks (Li et al., 2020; Garg and Ramakrishnan, 2020) Therefore, we use methods that can defend *candidate-agnostic* adversarial attacks as our baselines:

Adv-Train (HotFlip): Ebrahimi et al. (2017) introduces the adversarial training method used in defending against substitution-based adversarial attacks in NLP. It uses gradients to find actual adversaries in the embedding space.

Virtual-Adv-Train (TAVAT): Token-Aware VAT (Li and Qiu, 2020) use virtual adversaries (Zhu et al., 2019) to improve the performances in fine-tuning pre-trained models, which can also be used to deal with substitute-agnostic attacks. We follow the standard TAVAT training process to re-implement the defense results.

Further, there are some works that require candidate list, it is not a fair comparison with candidate-agnostic defense methods, so we list them separately:

[3]https://github.com/huggingface/transformers

Methods	Origin	Textfooler(K=12)	BERT-Atk(K=12)	Textfooler(K=48)	BERT-Atk(K=48)
IMDB					
BERT	94.1	20.4	18.5	2.8	3.2
RoBERTa	97.3	26.3	24.5	25.6	23.0
Adv-HotFlip (BERT)	95.1	36.1	34.2	8.1	6.2
TAVAT (BERT)	96.0	30.2	30.4	7.3	2.3
Rebuild & Ensemble (BERT)	93.0	**81.5**	**76.7**	**51.5**	**44.5**
Rebuild & Ensemble (RoBERTa)	96.1	**84.2**	**82.0**	**55.3**	**52.2**
AG's News					
BERT	92.0	32.8	34.3	19.4	14.1
RoBERTa	90.1	29.5	30.4	17.9	13.0
Adv-HotFlip (BERT)	91.2	35.3	34.1	18.2	8.5
TAVAT (BERT)	90.5	40.1	34.2	20.1	8.5
Rebuild & Ensemble (BERT)	90.6	**61.5**	**49.7**	**34.9**	**22.5**
Rebuild & Ensemble (RoBERTa)	90.8	**59.1**	**41.2**	**34.2**	**19.5**

Table 1: After-Attack Accuracy compared with defense methods that can defend *candidate-agnostic* attacks.

Methods	Origin	Textfooler(K=48)	Generic
IMDB			
BERT	94.0	2.0	45.0
Augmentation	93.0	18.0	53.0
ADA	93.5	17.0	-
ASCC	77.0	-	71.0
R & E	93.0	**52.0**	**79.0**

Table 2: After-Attack Accuracy compared with previous access-candidates methods based on BERT model. - means that the results are not reported in the corresponding papers.

Adv-Augmentation: We generate adversarial examples of the training dataset as a data augmentation method. We mix the generated adversarial examples and the original training dataset to train a model in a standard fine-tuning process.

ASCC: Dong et al. (2021) also use a convex-hull concept based on the candidate vocabulary as strong adversarial defense.

ADA: Si et al. (2020) use a mixup-strategy based on the generated adversarial examples to achieve adversarial defense.

4.4 Implementations

We use BERT-BASE and RoBERTa-BASE models based on the Huggingface Transformers [4]. We modify the virtual adversarial training process based on the implementation of FreeLB [5] and TAVAT [6]. The adversarial training hyper-parameters we use is different from FreeLB and TAVAT, since we aim to find large perturbations to simulate adversaries. We set adversarial learning rate α 1e-1 to and normalization boundary ϵ 2e-1 in all tasks. The ensemble size we use is $N = 16$ for all tasks and we will discuss the selection of N in the later section.

We use the TextAttack toolkit as well as the official code to implement adversarial attack methods [7] (Morris et al., 2020a). The similarity thresholds are the main factors of the attacking algorithm. We tune the USE (Cer et al., 2018) constraint 0.5 for the AG task and 0.7 for the IMDB task and 0.5 for the cosine-similarity threshold of the synonyms embedding (Mrkšić et al., 2016) which can re-produce the results of the attacking methods reported.

4.5 Results

As seen in Table 1, the proposed **Rebuild and Ensemble** framework can successfully defend strong attack methods. The accuracy of our defensing method under attack is significantly higher than no-defense models. Compared with previous defense methods, our proposed method can achieve higher defense accuracy in both IMDB task and AG's News task. The HotFlip and the TAVAT method are effective but not enough, which indicates that gradient-based adversaries are not very similar with actual substitutions. We can see that HotFlip and TAVAT methods achieve similar results which indicates that gradient-based adversarial training methods have similar defense ability no matter the adversaries are virtual or real since they are both unaware of the attacker's candidate list.

Also, the original accuracy (on the clean data) of our method is only a little lower than the baseline methods, which indicates that the defensive

[4] https://github.com/huggingface/transformers
[5] https://github.com/zhuchen03/FreeLB
[6] https://github.com/LinyangLee/Token-Aware-VAT
[7] https://github.com/QData/TextAttack

Different Settings of R & E					Origin	Textfooler(K=12)	BERT-Atk(K=12)
Train		Inference					
Joint	VAT	Ensemble	Rebuild	Insert			
Rebuild and Ensemble Method							
✓	✓	✓	✓	✓	93.0	86.0	77.0
Rebuild Train		No Ensemble					
✓	✓		✓	✓	93.0	63.0	52.0
✓	✓		✓		93.0	42.0	29.0
✓			✓	✓	95.0	45.0	34.0
✓			✓		95.0	29.0	17.0
		Inference Only					
		✓	✓	✓	94.0	72.0	60.0
			✓	✓	87.0	20.0	13.0
			✓		92.0	11.0	3.0
		✓			96.0	75.0	62.0
Baseline							
-	-	-	-	-	93.0	20.0	18.0

Table 3: Ablations results tested on attacking the IMDB task based on BERT models.

rebuild and ensemble strategy does not hurt the performances. The Roberta model also shows robustness using both original fine-tuned model and our defensive framework, which indicates our defending strategy can be used in various pre-trained language models.

Further, the candidate size is extremely important in defending adversarial attacks, when the candidate size is smaller, exemplified by $K = 12$, our method can achieve very promising results. As pointed out by Morris et al. (2020b), the candidate size should not be too large that the quality of the adversarial examples is largely damaged.

As seen in Table 2, we compare our method with previous access-candidates defense methods. When defending against the widely used Textfooler attack and Generic attack (Alzantot et al., 2018), our method can achieve similar accuracy even compared with known-candidates defense methods. As seen, data augmentation method cannot significantly improve model robustness since the candidates can be very diversified, using generated adversarial samples as an augmentation strategy does not guarantee robustness against greedy-searched methods like Textfooler and BERT-Attack.

4.6 Analysis

4.6.1 Ablations

We run extensive ablation experiments to explore the working mechanism in defending adversaries. We run ablations in two parts: (1) using the rebuild-trained model; (2) using the ensemble inference without training the model specifically.

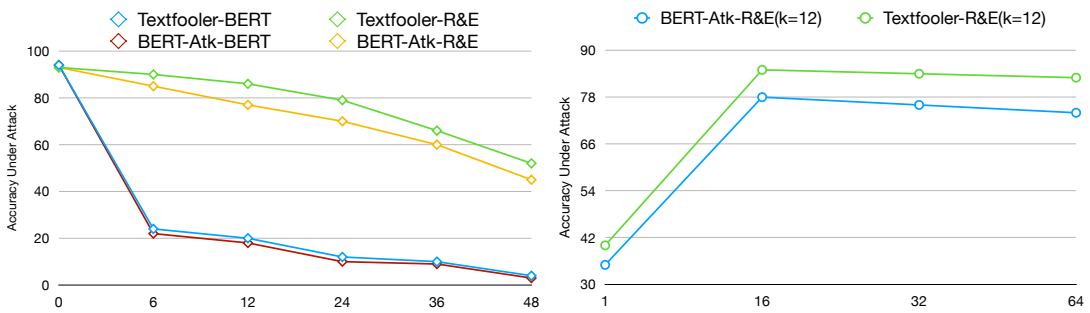
Firstly, we test the model robustness without using ensemble inference, that is, during inference, the ensemble size N is 1: We explore the effectiveness of incorporating the gradient-noise rebuild process. Also, we test the result of using the mask and rebuild strategy as well as the insert and rebuild strategy. Then we test the inference process: We use the fine-tuned model and the original masked language model as the prediction model and the rebuild model to run inference. We test the effectiveness of making multiple copies of rebuilt texts; We also explore how the two operations: mask and insert work during inference; Further, we setup an experiment using the noisy texts without the rebuild process.

As seen in the Table 3, we could explore the working mechanism in defending against the *candidate-agnostic* attacks via extensive results.

The observations indicate that:

(a) Rebuild Train is effective: The process in rebuild training allows the trained model to be aware of both the missing texts that need rebuilding and the classification labels of the inputs, which is helpful in rebuilding classification-aware texts. Without the rebuild trained model, the accuracy is even lower when rebuilding with the original masked language model during ensemble inference. However, rebuilding using the original MLM is not very much helpful, which indicates that the model trained with re-building process is important.

(b) Ensemble during inference is important: As seen, with the ensemble strategy, even random masking with an ensemble process can be helpful.



(a) Candidate-Size Influence (b) Ensemble Size Influence

Figure 3: Hyper-Parameter Selection Analysis

(c) Gradient-Noise Rebuild is helpful: without the gradient-noise rebuild process, the model can still defend adversaries.

4.6.2 Candidate Size Analysis

One key problem is that these attacking algorithms use a very large candidate size with a default set to around 50, which seriously harm the quality of the input texts. Therefore, we run experiments using different candidate size of these attacking algorithms to see how our defense strategy performs.

As seen in Fig. 3 (a), when the candidate is 0, the accuracy is high on the clean samples. When the candidate is 6, the normal fine-tuned BERT model cannot correctly predict the generated adversarial examples. This indicates that normal fine-tuned BERT is not robust even when the candidate size is small. While our approach can tolerate these limited candidate size attacks. When the candidate size grows, the performances of our defense framework drop by a relatively large margin. We assume that large candidate size would seriously harm the semantics which is also explored in Morris et al. (2020b), while these adversaries cannot be well evaluated even using human-evvaluations since the change rate is still low.

4.6.3 Ensemble Strategy Analysis

One key problem is that how many copies we should use in the rebuilding process, since during inference, it is also important to maintain high efficiency. We use two attack methods with $K = 12$ to test how the accuracy varies when using different ensemble size N.

As seen in Fig. 3 (b), the ensemble size is actually not a key factor. Larger ensemble size would not result in further improvements. We assume that larger ensemble size will *smooth* the output score which will benefit the attack algorithm. When the number of rebuild is not large, the inference efficiency is bearable.

Methods	Origin	Textfooler (K=12)
BERT	94.0	20.0
R & E (Mean)	93.0	82.0
R & E (Mean)(N=1)	93.0	42.0
R & E (Vote)	93.0	88.0
R & E (Vote)(N=1)	93.0	62.0

Table 4: Exploring the Ensemble Strategy

Further, we found that the ensemble strategy could use a voting mechanism to construct a *virtual score* as the final output. That is, the argmax votes can be used to craft a confident score. When the ensemble size $N = 1$, this process is a hard-score attack that only gives 1 and 0 as the output.

As seen in Table 4, the defensive result using the voting strategy is higher than using the average logits. So we can assume that incorporating our rebuild and ensemble strategy with output-score-hiding strategies could further improve the model robustness.

5 Conclusion and Future Work

In this paper, we introduce a novel rebuild and ensemble defense strategy against current strong adversarial attacks. The rebuild trained model can improve the model robustness since it maintains more semantic information while it also introduces a rebuild text process. The ensemble inference is also effective indicating that the multiple rebuilt texts are better than one. Experiments show that these proposed components can work coordinately to achieve strong defense performances. We are hoping such a defense process can provide hints for future works on adversarial defenses.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *CoRR*, abs/1804.07998.

Nicholas Carlini and David A. Wagner. 2016. Towards evaluating the robustness of neural networks. *CoRR*, abs/1608.04644.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. *arXiv preprint arXiv:1906.02443*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Xinshuai Dong, Hong Liu, Rongrong Ji, and Anh Tuan Luu. 2021. Towards robustness against natural language word substitutions. In *International Conference on Learning Representations*.

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

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. *arXiv preprint arXiv:1909.01492*.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. *CoRR*, abs/1909.00986.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is BERT really robust? natural language attack on text classification and entailment. *CoRR*, abs/1907.11932.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Linyang Li and Xipeng Qiu. 2020. Textat: Adversarial training for natural language understanding with token-level perturbation. *arXiv preprint arXiv:2004.14543*.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards deep learning models resistant to adversarial attacks.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2016. Virtual adversarial training for semi-supervised text classification. *ArXiv*, abs/1605.07725.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020a. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

John X. Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020b. Reevaluating adversarial examples in natural language. In *ArXiv*, volume abs/2004.14174.

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

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. 2018. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *CoRR*, abs/1805.06605.

Zhouxing Shi, Minlie Huang, Ting Yao, and Jingfang Xu. 2019. Robustness to modification with shared words in paraphrase identification. *CoRR*, abs/1909.02560.

Chenglei Si, Zhengyan Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2020. Better robustness by more coverage: Adversarial training with mixup augmentation for robust fine-tuning. *arXiv preprint arXiv:2012.15699*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2017. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*.

Jin Yong Yoo, John X. Morris, Eli Lifland, and Yanjun Qi. 2020. Searching for a search method: Benchmarking search algorithms for generating nlp adversarial examples. *ArXiv*, abs/2009.06368.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-wei Chang, and Xuanjing Huang. 2020. Defense against adversarial attacks in nlp via dirichlet neighborhood ensemble. *arXiv preprint arXiv:2006.11627*.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4904–4913, Hong Kong, China. Association for Computational Linguistics.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Thomas Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for language understanding. *arXiv preprint arXiv:1909.11764*.