Hide Your Weaknesses from Attackers: A Defense Method against Black-Box Adversarial Text Attacks

Kun Li[1,2] , Minxuan Lv[1,2] and Wei Zhou[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences
{likun2,lvminxuan,zhouwei}@iie.ac.cn

Abstract. Despite the significant successes of LLMs in generative tasks, the current preference in classification scenarios predominantly leans towards the use of pre-trained language models(PLM), taking into account the balance between cost and effectiveness. However, these models are prone to manipulation through black-box adversarial text attacks, where attackers modify texts in subtle ways to deceive the models. Typically, attackers follow a two-step process: first, they identify crucial sentence elements for the model, then they alter these elements by replacing, deleting, or adding words or characters.Previous research has mainly focused on creating adversarial examples for training, aiming to improve model resilience. These efforts often overlook defenses against the initial phase of identifying vulnerable targets. This paper introduces a defensive strategy against these first-stage attacks, leveraging concepts from differential privacy. We propose a novel approach, **Mask Regeneration**, which conceals the targets using a [Mask] token and employs a Mask Language Model (MLM) to generate misleading samples. Additionally, we observe that key targets often align with high attention values in the model. Based on this insight, we introduce an **Attention Shuffle** tactic, which randomizes the top-k attention values at each transformer layer, further disorienting attackers.The experiment shows that our defense method achieves better robustness gains than the State-of-the-art under three strong adversarial attacks for three typical NLP tasks, like sentiment analysis, textual entailment, and topic classification. Moreover, it is also demonstrated that the attack cost significantly increases when attacking our defense model.

Keywords: Natural language processing, Adversarial attack, Adversarial defense

1 Introduction

Deep Neural Networks (DNNs) have achieved remarkable success in the fields of machine learning and natural language processing (NLP). Notably, transformer-based pre-trained language models (PLMs) have demonstrated exceptional performance on diverse NLP tasks, including text classification, question answering, and reading comprehension. Nevertheless, researchers have discovered that NLP models based on PLMs exhibit significant vulnerability to adversarial attacks, resulting in substantial changes to the model's predicted outcomes.This paper is dedicated to the defense against char-level and word-level black-box adversarial text attacks. In contrast to white-box attacks, black-box attacks can solely access the output of victim models, making them closer to the actual attack scenario and presenting a greater challenge. The majority of black-box attacks involve two stages to achieve a successful attack, as illustrated in **Fig 1**. During the first stage, attackers attempt to identify the most critical targets based on the feedback received from victim models. In the second stage, attackers manipulate these vulnerable targets by replacing, deleting, or modifying them until a change in the model's output occurs.

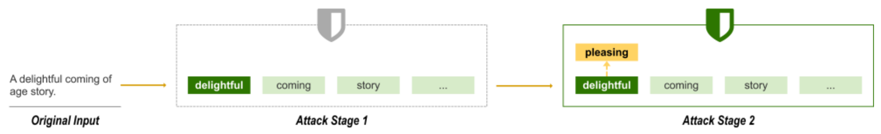

Fig. 1. A figure caption is always placed below the illustration. Short captions are centered, while long ones are justified. The macro button chooses the correct format automatically.

Several works have been conducted to defend against black-box attacks in NLP. [15] apply a gradient-based adversarial training approach, commonly used in the computer vision (CV) domain, to enhance the adversarial robustness of the text classification model. In another study [20] , the construction of a synonym map function is proposed to mitigate synonym replacement attacks. Additionally, several works utilize data augmentation techniques [22,18] and certification training methods [8] to enhance the robustness of the victim model against character/word perturbations in input text. Despite the critical role of the first stage in defense, most previous research has primarily focused on constructing more robust defense models, neglecting the development of specific methods for model protection during the initial attack stage. This paper presents a novel method that focuses on defending against black-box attacks, particularly during the initial attack stage. We propose two strategies: **Mask Regeneration** and **Attention Shuffle**. In the Mask Regeneration strategy, we observe that most black-box attacks employ a heuristic search strategy to identify vulnerabilities. These attacks attempt to substitute specific tokens with meaningless tokens to assess their significance by observing changes in the victim's output. Drawing inspiration from the differential privacy method, we propose the Mask Regeneration strategy, which involves replacing a greater number of tokens with [MASK] and utilizing the Mask Language Model (MLM) to restore these masked tokens. Consequently, the model's output remains largely unaffected, including these crucial tokens, regardless of the attacker's attempts to mask them. Additionally, through an analysis of adversarial data generated by attackers, we observed that vulnerable targets typically exhibit higher self-attention values in the transformer-based model. Consequently, we propose the Attention Shuffle strategy, which involves dynamically shuffling the attention values of these targets at each layer of the transformer to obfuscate them. Experiments show that we can mislead attackers and enhance the protection of the model by implementing these two strategies.

2 Related Work

In this section, we provide a brief overview of the black-box text adversarial attacks and defense methods.

2.1 Black-box Attack

Black-box text adversarial attacks include three types according to perturbations units: char-level attack, word-level attack, and sentence-level attack. Char-level attacks try to modify several characters in words to generate adversarial examples and fool the victim model. [4] proposed standard char-level attack methods, which all use a heuristic searching strategy to find essential words and then add imperceptible char-level perturbations to the selected words through swapping, flipping, deletion, and insertion characters.Word-level attacks manipulate the whole word rather than several characters in words, which is more imperceptible to humans and very hard to detect by models. PWWS[17] is one of the most effective word-level methods, which uses some synonyms to replace vulnerable words. Based on previous methods, [10] add the semantic constraint on synonyms choosing to ensure the generated adversarial samples are semantically stable and more readable. Not surprisingly, two attack methods[5,14] use pre-trained MLM to generate replacement words rather than use synonyms directly.Recently, [7] proposed a two-stage attack process, including victim model extract and adversarial example transfer. In this paper, we evaluate our defense model against three competitive attackers mentioned above, including TextBugger, TextFooler, and PWWS.

2.2 Black-box Defense

The rapid development of black-box adversarial text attackers requires effective defense methods to fight against the threats. We summarize two defense methods: adversarial example restore and model robustness enhancement, the former attempts to restore the adversarial samples created by attackers to some normal samples. [16] proposed a spelling check model to distinguish and recover adversarial examples in the char-level. [26] train a discriminator and estimator module to discriminate both char-level and word-level perturbations. [20] and [11] constructed a robust reprocess function which maps all typos and synonyms to the same feature representation to reduce the attack effect. There are more defense methods designed for enhance model robustness through adversarial training. [15] try to use some very successful adversarial training methods like FGSM in CV domain to build a defense model of text. [21] want to train a more robust model through data augmentation approaches. However, training adversarial examples produced by text augmentation are extremely insufficient that can not cover all attack scenarios. So [18] proposed a new method to cover a much more significant proportion of the attack search space. All of the above defense methods can improve the model's defense, but few of them are designed from the attackers' perspective, which is the main improvement of our paper.

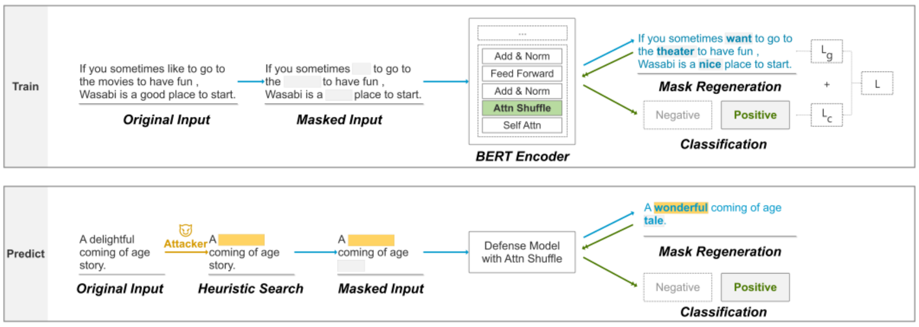
Train
If you sometimes like to go to the movies to have fun , Wasabi is a good place to start.
Original Input
If you sometimes to go to the to have fun , Wasabi is a place to start.
Masked Input
...
Add & Norm
Feed Forward
Add & Norm
Attn Shuffle
Self Attn
BERT Encoder
If you sometimes want to go to the theater to have fun , Wasabi is a nice place to start.
Mask Regeneration
Negative
Positive
Classification
L_g
+
L_c
L
Predict
A delightful coming of age story.
Original Input
Attacker
A coming of age story.
Heuristic Search
A coming of age
Masked Input
Defense Model with Attn Shuffle
A wonderful coming of age tale.
Mask Regeneration
Negative
Positive
Classification

Fig. 2. Overview of our defense model. we train a BERT model with regenerated samples, then regenerate sentences and obscure key tokens to perturb the attack..

3 Methodology

3.1 Problem Formulation

We first give some formula descriptions about black-box adversarial attacks and defense models.A black-box attacker aims at adding some perturbations ϵ to the n words input sequence $S = \{w_1, w_2, \dots, w_n\}$ to create adversarial sample S'according to the prediction results of victim model F.Such that for the victim model, $F(x) = y$ and $F(x') = y'(y \neq y')$, and the perturbed adversarial sample should be imperceptible to humans. A defense model D should tolerate the adversarial attacks and output the correct predictions, i.e. $F(x') = y$. When the defense model itself acts like a victim model and faces attacks directly, it can also significantly reduce the attack success rate and increase the attack cost.

3.2 General Architecture

As mentioned in the first section, our paper introduces a new defense method mainly works at the first attack stage. Our approach includes two strategies: Mask Regeneration and Attention-Shuffle. **Fig 2** shows an overview of our defense method, including model training and prediction. We can see that every input sample will be randomly masked, then use a pre-trained mask language model, such as BERT, to regenerate tokens and build a new sample for training and predicting.During training, we optimize the loss of the re-generation task and the classification task, and a hyper-parameter λ is used to control the weight. Additionally, the generated sample is input into the Attention-Shuffle encoder to obtain features and make classification predictions. The attention-Shuffle strategy perturbs the self-attention in a transformer-based encoder. Specifically, the top k attention values are randomly shuffled. We can alleviate the model bias on some important but vulnerable tokens. Next, the implementation and principles of the two strategies will be introduced in detail.

3.3 Mask Regeneration

The Mask Regeneration strategy works at both training and predicting stages. At the training stage, every original input sentence is mapped to another generated sentence by the mask language model. These regenerated new samples are good data augmentation for training to improve robustness of model. At the predicting stage, when an attacker tries to mask tokens for vulnerable token heuristic searching, the regenerate strategy will restore them to a regenerated sample. This regenerated obfuscation makes it difficult for attackers to find valuable targets. At the same time, since many data regenerated by the MLM is used for training, it can be guaranteed that these new samples will hardly affect the predicting results. Specifically, for a input sample $x = \{w_1, w_2, \dots, w_n\}$, we randomly mask 15 percent tokens (exclusion [mask] itself) as $x_{mask} = \{w_1, [mask], .., [mask], .., w_n\}$. Then **all** tokens will be Regenerated through a standard MLM process by the pre-trained language model.Since generating text with discrete decoding steps(*argmax*) will be used in the classification task, the joint architecture is not differentiable for gradient back propagation. So we choose to use Gumble-softmax [9] function to replace *argmax* function.Instead of sampling a word from the vocabulary directly by *argmax*, we calculate a distribution vector $Z_k = \{z_1, z_2, .., z_i, .., z_m\}$ of k-th prediction words in the input sentence to represent the probability of choosing the i-th word overall *m* words of vocabulary.

$$z_i = \frac{exp\left(\frac{log(p_i) + g_i}{\tau}\right)}{\sum_{j=1}^{m} exp\left(\frac{log(p_j) + g_i}{\tau}\right)} \tag{1}$$

where p_i is the MLM predicition result of i-th word, τ is the temperature hyper-parameter, g_i is i.i.d sample drawn from $Gambel(0,1)$.

$$g_i = -log(-log(\mu_i)), \mu_i \sim u(0,1) \tag{2}$$

Like *argmax* function, the probability of the generated word is close to one and other words close to zero in distribution Z_k. So We can obtain the Regenerate text embedding $E_{x^*} = \{e_1, e_2, .. e_k .., e_n\}$ by multiplying Z_k. with the word embedding matrix E of pre-trained language model.

$$e_k = Z_k * E \tag{3}$$

Then, the generated embedding E_{x^*} is used for downstream classification task through the *Attention Shuffle* encoder.In training stage, both MLM task and downstream classification task are optimized, including L_g and L_c.

$$L_g = -logp_c(x|x_{mask}, \theta) \tag{4}$$

$$L_c = -logp_g(y|E_{x^*}, \theta) \tag{5}$$

Notice that p_c and p_g **share weights** except a linear layer for classification. Finally, we optimize the L_g and L_c at the same time as below:

$$L = \lambda \cdot L_g + (1 - \lambda) \cdot L_c \tag{6}$$

where λ controls the weight of L_g and L_c.

3.4 Attention Shuffle

The attention-Shuffle strategy tries to randomly shuffle the top k(5 in practice) highest attention values to alleviate the model bias on some tokens. Specifically, we add a shuffling process to the standard BERT encoder. We first find the top k attentions values and indices for every token except themselves. Then we shuffle the values of these indices as shown in **Fig 3**. These shuffled attention values will be used to calculate the hidden states of input, then put to the next layer, and finally used to predict classification labels.

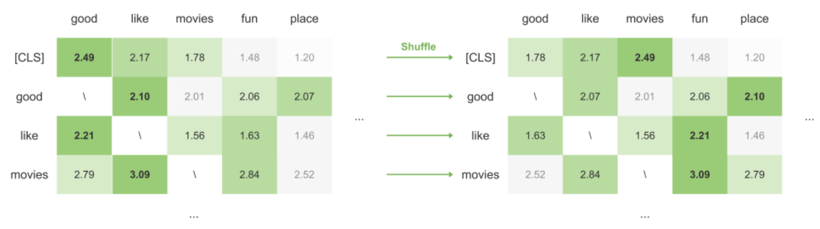

Fig. 3. An example of an attention shuffle inside the encoder layer. The attention value of the most concerned token could be shuffled to others.

4 Experiment

4.1 Experiment Setups

Targeted Attack Evaluation. [18] introduced two different ways to evaluate defense effectiveness under adversarial attacks used in previous works: Static Attack Evaluation(SAE) and Targeted Attack Evaluation (TAE). SAE evaluates the defense model on the adversarial samples derived from original victim model. A lot of previous work has used this experimental setup [17,20]. However, SAE does not correspond to real scenarios, and it cannot evaluate the performance of the defense model itself as the victim face attacks. So, in this paper, we choose to use the TAE setup in all experiments which try to Regenerate a new set of adversarial examples when every defense model is being evaluated.

Datasets. We evaluate our proposed defense method on three different datasets.SST-2 [19] is a binary sentiment classification dataset which contains 1821 test samples; *Agnews* [25], a multi-class topic classification dataset, include 10 classes. And a natural language inference dataset *SNLI* [1]. For Agnews and SNLI, it is prolonged to attack the whole test set (about 10K samples) using three different attackers. So we take 2k subset as the test samples for attack evaluation.

Attackers. In order to better examine the effect of our defense model, we use three different competitive black-box adversarial attackers including two word-level *PWWS* [17], *TextFooler* [10] and one char-level *TextBugger* [13] as our attack models in main experiments. Both attackers can access model prediction scores but not gradients or other victim information.In all experiments, we use the OpenAttack module [24] to build all these attackers efficiently.

Baselines. We compare our defense model with five baselines as follows, which can apply to the same experimental setting. *FGSM*, a adversarial training method, introduced by [6], then [15] apply it to NLP domain. *SEM* [20] is an adversarial example restore method that maps all synonyms to the same feature representation to defend against word-level attacks. *RSE* [22], a simple but effective synonyms text augmentation based defense method. AMDA [18], a new data augmentation-based defense method that tries to linearly interpolates the representations of pairs of training samples to form new virtual samples.*HiddenCut* [2] cuts off some hidden embeddings according to attentions randomly to build more generative model.

Implementation Details. Our experiment uses the BERT-Base-Uncased [3] as our pre-trained language model with 12-layer transformer blocks, 768-dimension hidden state, 12 attention heads, and total 110M parameters. In practice, we implement the BERT encoder based on huggingface transformers module [23]. Considering the average length of sentences in three datasets, we set the max sequence length to 128. At the training stage, we adopt Adam optimizer [12] and set the learning rate to be 5e-5, with β1=0.9,β2=0.999. Because we use a single Tesla V100-16G GPU to train and attack evaluate, the batch size is set to be 32 in all our experiments. Loss weight λ of the generation task and prediction task is set to 0.5.

Table 1. Accuracy of various baselines and our defense methods against attackers, including TF(TextFooler), TB(TextBugger), PWWS. NA (No Attack) is the original accuracy without attacks. REG is the model that only uses the Mask Regeneration strategy, and AS is the model that only replaces the BERT encoder with the attention shuffle enhanced BERT encoder. REG + AS is our whole defense method.The best performance for defense methods under each attack is **boldfaced**, and the second-best performance is underlined.

	SST2				AgNews				SNLI			
	NA	TF	TB	PWWS	NA	TF	TB	PWWS	NA	TF	TB	PWWS
ORIG	**93.2**	32.5	23.1	34.1	**94.5**	36.6	12.3	40.7	85.4	14.1	11.3	13.6
FGDM	89.3	38.7	23.7	34.2	94.2	40.2	20.3	46.3	85.1	15.5	12.5	15.1
MixUp	91.7	36.4	26.6	40.5	93.3	40.7	13.4	41.8	85.4	14.7	13.1	16.3
SEM	87.9	44.3	34.5	48.5	94.1	57.2	14.5	70.3	**86.3**	20.3	11.8	20.3
RSE	91.3	38.4	27.6	42.4	94.1	55.2	21.2	59.7	85.3	22.8	11.7	22.5
HiddenCut	89.4	60.7	55.6	59.1	94.4	77.8	65.1	78.1	78.2	39.2	35.8	34.9
REG	89.1	68.9	**63.4**	63.6	93.5	**85.7**	74.2	82.7	73.4	47.7	48.9	45.5
AS	90.7	52.3	44.6	48.4	93.8	54.2	30.1	56.8	82.9	24.9	21.2	24.9
REG+AS	87.8	**70.7**	62.2	**64.5**	93.6	85.5	**75.6**	**84.3**	72.6	**48.3**	**49.1**	**46.3**

4.2 Main Results Analysis

Table 1 reports the main result of our method, which includes both clean accuracy (Not Attack) and defense accuracy under three attack algorithms (TextFooler, TextBugger, and PWWS) on three different datasets compared to six baselines. We use *ORIG* to represent the adversarial accuracy of the original BERT model trained without using any defense method. We use *REG* to represent the result of the defense model only with the Mask Regeneration strategy, *AS* to represent the defense model only with the Attention-Shuffle strategy, and *REG+AS* is our whole defense method. We can observe that: (1) the original BERT model without any defense mechanism achieves the best clean accuracy. We think the defense performance and clean accuracy are a trade-off. It is hard to improve them at the same time. (2) Attention-Shuffle strategy can achieve a fair defense accuracy compared to the original BERT model and many baselines. It is an ablation experiment that proves the validity of *AS* strategy. (3) our method outperforms almost all baselines on the defense accuracy across the three different datasets under all the attack algorithms.Then (*REG+AS*) can achieve further defense improvement compared to *REG* in most cases. This proves that *REG* and *AS* can better complement each other to improve model robustness under various adversarial attacks.(4) According to the result on the SNLI dataset,the natural language inference task seems very vulnerable to attacks. Most baselines are defenseless, and accuracy dropped a lot under attacks. Our method has a fair defense result but significantly decreases the original accuracy. We think the NLI sample includes premise and hypothesis, and their semantic relationship determines the final inference result. Mask Regeneration strategy may affect these collections heavily. (5) TextBugger is the most strong attacker for many baselines, especially to the word substitution-based defense method. It makes a huge accuracy drop in both three tasks. Our method can work well against char-level and word-level adversarial attacks.

Table 2. The perturbation rate of adversarial examples at SST-2 dataset after a successful attack to different defend methods. LED represents the levenshtein distance and WMR represents the word modify rate, higher is better.

	TF		PWWS	
	LED	WMR	LED	WMR
ORIG	4.99	27.2	3.95	22.1
FGDM	5.02	27.8	4.21	24.3
MixUp	5.16	28.5	4.36	24.4
SEM	5.01	28.2	4.04	23.9
RSE	5.25	28.6	4.16	24.2
HiddenCut	5.27	29.8	4.83	27.9
REG	5.51	**31.3**	5.27	30.1
AS	5.19	28.4	4.38	25.5
REG+AS	**5.62**	30.9	**5.31**	**30.6**

4.3 Attack Cost Analysis

As described in the introduction section, our approach aims to mislead attackers to unimportant parts rather than some meaningful tokens, thus increasing the attack cost and reducing the attack success rate. **Table 2** shows the final perturbation rate of adversarial examples at SST-2 dataset after a successful attack. We find that our defense models incur higher word modification rates under attacks. A higher perturbation rate means higher attack costs, which are more likely to make the crafted adversarial examples incomprehensible to humans. For example, PWWS only needed to modify 22% of the words to complete the attack on the original model but needed to modify 30% on our defense model. On the other hand, we also evaluate the attack cost directly by total victim query times. The results are shown in **Table 3**. Victim query times are the average number of accesses to the victim model required for a successful attack. We can see that attackers need to query more times to our defense model more to find the weakness compared to the original model and other baselines. All the current attack algorithms are very inefficient. If we can further increase the attack cost, we can also make the attack fail, thus improving the defense capability of models.

Table 3. Performance on total victim query times at SST-2 dataset, higher is better.

	TF	TB	PWWS
Original	62.3	51.4	114.6
FGDM	62.7	51.6	114.9
MixUp	62.4	57.5	115.3
SEM	65.2	57.4	115.2
RSE	63.6	55.3	115.5
HiddenCut	71.8	60.3	115.7
REG	73.2	62.9	116.1
AS	68.4	59.1	115.5
REG+AS	75.3	64.2	116.3

4.4 Disturb Attack Targets

One of the primary purposes of our defense approach is to disturb the attacker's choice of targets. In **Fig 4**, We choose four sentences from the sentiment analysis dataset SST-2 to further reveal the effect of our defense method. When the original model without the defense method is attacked, attackers easily find the most important word as a target. However, our defense method hides most of these essential words in low order. This causes the attacker to make many invalid attack attempts.

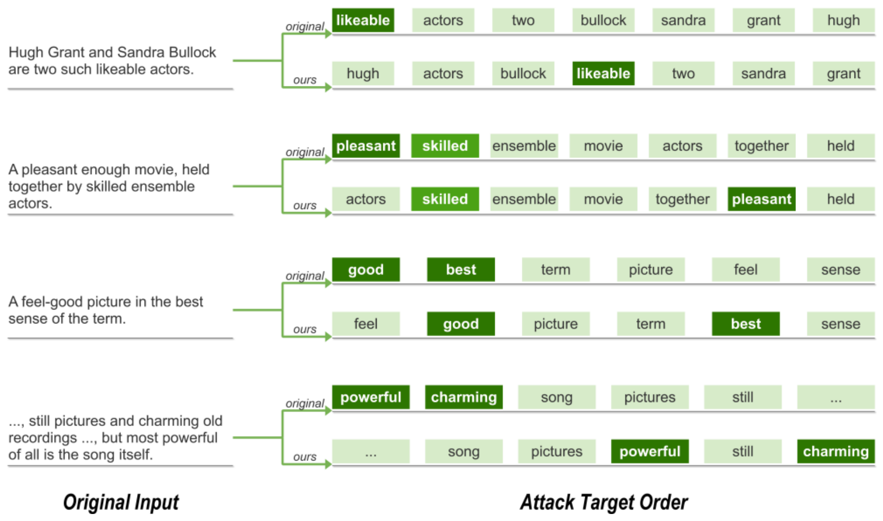
Hugh Grant and Sandra Bullock are two such likeable actors.
original
likeable actors two bullock sandra grant hugh
ours
hugh actors bullock likeable two sandra grant
A pleasant enough movie, held together by skilled ensemble actors.
original
pleasant skilled ensemble movie actors together held
ours
actors skilled ensemble movie together pleasant held
A feel-good picture in the best sense of the term.
original
good best term picture feel sense
ours
feel good picture term best sense
..., still pictures and charming old recordings ..., but most powerful of all is the song itself.
original
powerful charming song pictures still ...
ours
... song pictures powerful still charming
Original Input
Attack Target Order

Fig. 4. Examples of target selection order when attackers access different victim models on SST-2. Obviously, our defense method can hide some very meaningful words like 'best' or 'pleasant' in the sentiment analyse task.

5 Conclusion

In this paper, we propose a novel defense method against black-box attacks that primarily focuses on the first attack stage to redirect attackers toward less important components in a sentence. To achieve this defense objective, we introduce two strategies: Mask Regeneration and Attention Shuffle. The Mask Regeneration strategy involves mapping the original input sentence to a newly generated sentence, which effectively eliminates attack perturbations and confuses the attacker's selection of the target. The Attention Shuffle strategy randomly rearranges the top attention values to mitigate model bias towards specific tokens. Through extensive experimentation, we demonstrate a significant increase in the attack cost when targeting our defense model. In future work, we will concentrate on enhancing our performance in the natural language inference task.

6 References

1. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: EMNLP. pp. 632–642. The Association for Computational Linguistics (2015)
2. Chen, J., Shen, D., Chen, W., Yang, D.: Hiddencut: Simple data augmentation for natural language understanding with better generalizability. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4380–4390 (2021)
3. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1). pp. 4171–4186. Association for Computational Linguistics (2019)
4. Gao, J., Lanchantin, J., Soffa, M.L., Qi, Y.: Black-box generation of adversarial text sequences to evade deep learning classifiers. In: IEEE Symposium on Security and Privacy Workshops. pp. 50–56. IEEE Computer Society (2018)
5. Garg, S., Ramakrishnan, G.: BAE: bert-based adversarial examples for text classification. In: EMNLP (1). pp. 6174–6181. Association for Computational Linguistics (2020)
6. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (Poster) (2015)
7. He, X., Lyu, L., Sun, L., Xu, Q.: Model extraction and adversarial transferabil- ity, your BERT is vulnerable! In: NAACL-HLT. pp. 2006–2012. Association for Computational Linguistics (2021)
8. Huang, P., Stanforth, R., Welbl, J., Dyer, C., Yogatama, D., Gowal, S., Dvijotham, K., Kohli, P.: Achieving verified robustness to symbol substitutions via interval bound propagation. In: EMNLP/IJCNLP (1). pp. 4081–4091. Association for Computational Linguistics (2019)
9. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: ICLR (Poster). OpenReview.net (2017)
10. Jin, D., Jin, Z., Zhou, J.T., Szolovits, P.: Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In: AAAI. pp. 8018–8025. AAAI Press (2020)

11. Jones, E., Jia, R., Raghunathan, A., Liang, P.: Robust encodings: A framework for combating adversarial typos. In: ACL. pp. 2752–2765. Association for Computational Linguistics (2020)
12.
13. Jones, E., Jia, R., Raghunathan, A., Liang, P.: Robust encodings: A framework for combating adversarial typos. In: ACL. pp. 2752–2765. Association for Computational Linguistics (2020)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (Poster) (2015)
15. Li, J., Ji, S., Du, T., Li, B., Wang, T.: Textbugger: Generating adversarial text against real-world applications. In: NDSS. The Internet Society (2019)
16. Li, L., Ma, R., Guo, Q., Xue, X., Qiu, X.: BERT-ATTACK: adversarial attack against BERT using BERT. In: EMNLP (1). pp. 6193–6202. Association for Computational Linguistics (2020)
17. Miyato, T., Dai, A.M., Goodfellow, I.J.: Adversarial training methods for semi- supervised text classification. In: ICLR (Poster). OpenReview.net (2017)
18. Pruthi, D., Dhingra, B., Lipton, Z.C.: Combating adversarial misspellings with robust word recognition. In: ACL (1). pp. 5582–5591. Association for Computational Linguistics (2019)
19. Ren, S., Deng, Y., He, K., Che, W.: Generating natural language adversarial examples through probability weighted word saliency. In: ACL (1). pp. 1085–1097. Association for Computational Linguistics (2019)
20. Si, C., Zhang, Z., Qi, F., Liu, Z., Wang, Y., Liu, Q., Sun, M.: Better robustness by more coverage: Adversarial and mixup data augmentation for robust finetuning. In: ACL/IJCNLP (Findings). Findings of ACL, vol. ACL/IJCNLP 2021, pp. 1569–1576. Association for Computational Linguistics (2021)
21. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP. pp. 1631–1642. ACL (2013)
22. Wang, X., Jin, H., He, K.: Natural language adversarial attacks and defenses in word level. CoRR **abs/1909.06723** (2019)
23. Wang, Y., Bansal, M.: Robust machine comprehension models via adversarial training. In: NAACL-HLT (2). pp. 575–581. Association for Computational Linguistics (2018)
24. Wang, Z., Wang, H.: Defense of word-level adversarial attacks via random substitution encoding. In: KSEM (2). Lecture Notes in Computer Science, vol. 12275, pp. 312–324. Springer (2020)
25. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: EMNLP (Demos). pp. 38–45. Association for Computational Linguistics (2020)
26. Zeng, Guoyang, et al. "Openattack: An open-source textual adversarial attack toolkit." arXiv preprint arXiv:2009.09191 (2020).
27. Zhang, X., Zhao, J.J., LeCun, Y.: Character-level convolutional networks for text classification. In: NIPS. pp. 649–657 (2015)