Combating Word-level Adversarial Text with Robust Adversarial Training

Xiaohu Du[†,1], Jie Yu[†,1], Shasha Li[1], Zibo Yi[1], Hai Liu[*,2] and Jun Ma[1]
[1]College of Computer, National University of Defense Technology, Changsha, China
[2]Logistical Research Institute of Science and Technology, Beijing, China
{duxiaohu18, yj, shashali, yizibo14, hailiu, majun}@nudt.edu.cn

***Abstract*—NLP models perform well on many tasks, but they are also easy to be fooled by adversarial examples. A small perturbation can change the output of the deep neural network model. This kind of perturbation is hard to be perceived by humans, especially adversarial examples generated by word-level adversarial attack. Character-level adversarial attack can be defended by grammar detection and word recognition. The existing word-level textual adversarial attacks are based on synonym replacement, so adversarial texts usually have correct grammar and semantics. The defense of word-level adversarial attack is more challenging. In this paper, we propose a framework which is called Robust Adversarial Training (RAT) to defend against word-level adversarial attacks. RAT enhances the model by combining adversarial training and data perturbation during training. Our experiments on two datasets show that the model based on our framework can effectively defend against word-level adversarial attacks. Compared with the existing defense methods, the model trained under RAT has a higher defense success rate on 1000 adversarial examples. In addition, the accuracy of our model on the standard testing set is also better than the existing defense methods, and the accuracy is very close to or even higher than that of the standard model.**

***Index Terms*—Deep neural network, Deep learning, Adversarial training, Adversarial attack**

I. Introduction

Machine learning and deep learning technologies have achieved great success in some tasks, such as text classification and sentiment classification. However, some deep learning models have proven to be vulnerable to adversarial attack [1]. The adversarial attack is a minor modification of the input, but this modification which is not easy to be detected by humans can change the judgment of classifier. This modification is called perturbation, and the text generated by the attack is adversarial example. These adversarial examples expose some vulnerabilities of neural network classifier, which can be used to evaluate and improve deep learning models. Generally speaking, adversarial attack methods in text classification are mainly divided into three categories, they are character-level adversarial attack [2], word-level adversarial attack [3] and sentence-level adversarial attack [4]. In this paper, we mainly discuss word-level adversarial attacks, because they are imperceptible and difficult to defend. A word-level attack is that the attacker makes perturbations at the word level. These perturbations can be word replacement, deletion, and addition. The most effective method for word-level perturbation is synonym replacement. Table I shows the adversarial examples of word-level attacks. A word-level adversarial example can be a completely different synonym replacement to imitate different people's different ways of expressing the same thing, such as the first example. It also can be replaced with the same form but different tenses word to imitate people's accidental typing errors, as the second example in the table. The text after replacement has the same semantics with the original text, but it is easy to be misclassified by the deep learning model. How to avoid the misclassification of model on these adversarial examples has been a hot research topic in recent years.

TABLE I: The word-level adversarial examples.

text	label
I am sorry but this is the worst film.	Negative
I am sorry but this is the harshest film.	Positive
It deserves watch it now fantastic.	Positive
It deserve watch it now fantastic.	Negative

The existing defense ideas are basically derived from the generation method of adversarial examples. Since there are synonym replacement attacks, there are naturally synonym replacement defense methods. Wang et al. [5] propose a word-level defense method called Synonyms Encoding Method (SEM), which map all the synonyms to the unique encoding in the word space, so that when the model processes synonyms, the impact of synonym replacement can be eliminated. The robustness of the model is enhanced to some extent. However, the word diversity between different words are also deprived for the reason that the synonyms are mapped to the same word vector representation. The result is that the performance of the model on the testing set is not as good as the standard model. Wang et al. [6] propose a method called Random Substitution Encoding (RSE), which generates neighborhood examples coming from random synonym replacements, they use random synonym replacement for the entire text to obtain neighborhood examples, and try to eliminate the impact of synonym replacement at the sentence level. Like the SEM method, RSE performs well on defending adversarial ex-

This work is supported by the National Key Research and Development Program of China (No. 2018YFB0204301)

[†]These authors contributed equally to this work.

[*]Corresponding Author.

amples. However, RSE replaces all the text in the standard training set during the training process, which causes that the accuracy of the model trained by RSE is not as good as the standard model on the testing set.

It is a common problem that the accuracy of the testing set of the model is reduced. Existing study [7] has shown that models that perform poorly on the testing set usually have high adversarial robustness. Therefore, enhanced network model should be trained to have similar test performance. This standard ensure that defense method helps improve robustness. Robustness should not be at the expense of test accuracy.

In view of the drawbacks of the existing defense methods, we propose a novel defense framework called Robust Adversarial Training (RAT), which adds a word perturbation step in the training phase on the basis of adversarial training. The experiments show that our framework can effectively defend against word-level adversarial examples and outperforms the latest model.

In summary, our contributions in this work are as follows:

(1) We propose a defense framework called RAT, which can effectively identify adversarial examples and reduce the misclassification rate of the model in adversarial examples. The effectiveness of our method is verified in the combination of multiple deep neural network models, multiple datasets and multiple defense methods. Our defense framework outperforms the latest word-level defense methods.

(2) The model trained by RAT has better performance on the testing set than the existing methods, and its accuracy is close to or even higher than the accuracy of the standard model.

The rest of this article is organized as follows: In Section II, we describe related work done by previous scholars, focusing on word-level textual adversarial attacks and defenses, some of which will become our comparison objects. In Section III, we describe our defense method in detail. In Section IV, we introduce the relevant content of the experiment, including the target model, datasets, experimental parameters, and experimental results under different parameters. Section V introduces the final conclusion and future work.

II. Related Work

In 2014, Szegedy et al. [1] find that deep neural networks used for image classification can be deceived by images that have been added with tiny pixel perturbations. Experiments show that the image classifier has a high rate of misclassification, but humans have not detected this change in the image [8]. In recent years, researchers begin to pay attention to adversarial attack of text. For example, in 2017, Jia et al. [9] generate adversarial examples for deep neural networks in reading comprehension tasks. We take the whole text as the research object. For a given original text s, the trained deep neural model will give it a label with a certain degree of confidence. When an attacker uses a certain attack method to generate an adversarial example against target model, the target model may output the wrong label. Let y and $\widehat{y}$ denote

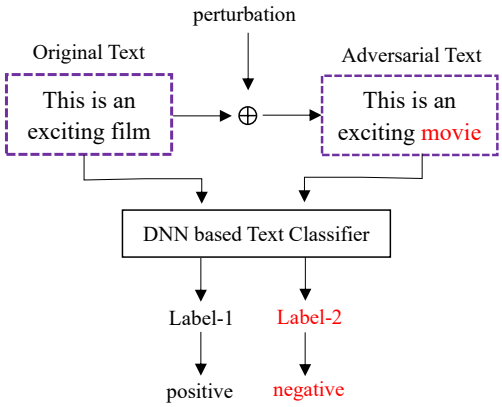


Fig. 1: The process of generating adversarial example.

the original and adversarial label. The adversarial example is generated by imperceptible perturbations. For a text of n words, $s = \{w_1, w_2, ..., w_n\}$, and a valid adversarial example $s^{'}$ should conform to the following requirements:

$$\widehat{y} \neq y, \ and \ Sim(w_i, w_i^{'}) \leq threshold \tag{1}$$

where Sim is the distance between w_i and $w_i^{'}$ in the word space. It should be less than a threshold. Figure 1 shows the process and results of the adversarial attack.

A. Adversarial Attack in Word Level

In 2018, Samanta et al. [10] propose a word-level black-box attack called Word Saliency (WS), which generates adversarial examples through operations such as deleting, replacing, and adding words. WS first calculates the contribution of each word to the classification result, and sorts them from largest to smallest. If a word has a large contribution and is an adverb, the word is deleted. The candidate words of each word are found in the remaining words. Finally, the candidate words that contribute least to the correct classification of the model are selected for replacement.

Alzantot et al. [3] introduce the group optimization algorithm to the textual adversarial attack for the first time, and propose a word-level black-box attack algorithm based on the Genetic Algorithm (GA), The first step of GA is Perturb. Perturb randomly selects a word and calculates the N neighbors of the word as candidate words according to the GloVe word space. Multiple uses of Perturb can generate the initial population. Individuals of each generation can obtain the prediction score of the corresponding target label by querying the attacked model. If there is an example that changes the target label in these examples , the optimization is completed. Otherwise, examples are drawn in pairs with a certain probability for crossover. The crossover process is that attacker randomly extracts words from the two examples in order to form new offspring. After one round, Perturb is used again for the second round of optimization until the model label changes.

Ren et al. [11] propose a black-box attack called Probability Weighted Word Saliency (PWWS) in 2019. PWWS is an improved method on the basis of the WS. It first calculates the word saliency vector $S(x)$ of each word w_i in the text x according to the WS method. When choosing priority replacement words, PWWS comprehensively calculates the degree of change in the classification probability after the replacement and the value of each word significance. Then it uses $x^* = (w_1, w_2 ... w_n)$ to indicate the text replacing w_i with w_i^*, and uses $\Delta P_i^* = F_Y(x) - F_Y(x_i^*)$ to indicate the significance of replacing w_i. Finally, the score of w_i defined by the following function:

$$H(x, x_i^*, w_i) = \phi(S(x))_i * \Delta P_i^* \tag{2}$$

Where $\phi(z)_i$ is the softmax function,

$$\phi(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \tag{3}$$

PWWS sorts all words in descending order based on H and selects candidate words, and then uses greedy search to traverse the entire candidate words until the model label changes. It is essentially a statistical method. The experiments on some datasets further reduce the accuracy of the model than WS. PWWS is also one of the main adversarial attacks in our experiments.

B. Defense of Adversarial Attack in Word Level

Adversarial training [1, 12] is the earliest defense method against adversarial examples. The basic idea is that defender uses various adversarial attack methods to make a set of large adversarial examples and add them to the training dataset. The training data includes a mixture of adversarial examples and corresponding original examples. Adversarial examples can be detected to a certain extent by this method. Adversarial training can be used to regularize deep neural networks, reduce overfitting, and improve the robustness of neural networks.

Wang et al. [5] propose a defense method called Synonym Encoding Method (SEM). It is easy to find almost all the neighbors of the input text. Based on this, SEM is proposed to locate the neighbor texts of the input texts. SEM assumes that the neighbor of the input texts is their synonymous texts. SEM uses synonyms to replace words in text to produce synonymous texts. A robust model will give synonymous texts the same label. To construct such a map, the synonyms need to be consolidated and assigned a unique encoding for each consolidated synonym. SEM creates and saves the word vector matrix of the synonym encoding dictionary, and then uses the synonym encoding word dictionary to train the CNN, LSTM and BI-LSTM models. Finally, in the experimental part, the model of retraining is attacked again. The results show that the model accuracy rate after using SEM is higher than that of normal training and adversarial training.

In 2020, Wang et al. [6] propose a defense method called Random Substitution Encoding (RSE). The main attack process is as follows: For the input text s, RSE randomly selects a replacement rate between the given maximum and minimum replacement rate, then generates a candidate word set C of the text and select synonyms for each word in C to get the perturbed text $s^{'}$. RSE replaces s with $s^{'}$ when training. In the testing phase, the adversarial examples from testing set are input the enhanced model to test the effect of RSE. The experiments show that the overall defense effect of RSE is better than SEM on three models of CNN, LSTM, and Bi-LSTM, and three datasets of IMDB, AG's News and Yahoo! Answers.

III. Methods

The purpose of the defense against the attack is that the model should show the same performance between original and adversarial text. The work we do on defense is essentially a method of data expansion. Our defense framework is designed to enhance the robustness of the model.

A. Motivation

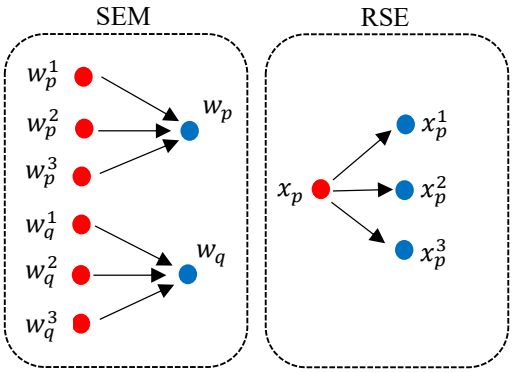


Fig. 2: Existing methods of defense against adversarial attack.

The existing methods of defense against adversarial attack have their limitations. As shown in Figure 2, SEM uses synonym encoding to defend against adversarial attacks. It encodes synonyms as the only encoding during training, so that the trained model has better robustness on the adversarial examples, but its shortcomings are also as shown in the figure. Because the synonyms are mapped to an encoding, the training examples are greatly reduced. The differences of synonyms in different texts are also deprived. The final result is that SEM performs poorly on the standard training set. RSE adopts the opposite defensive idea. It expands the training example by random word replacement for each text in all training sets. The advantage of this method is that it retains the features of the original data while adding more adversarial example features, but it also has shortcomings. Because all texts in the training set are replaced, and this replacement is completely random. There is no guarantee that the random replacement texts must contain the features of the adversarial examples, and they may be completely unrelated texts. The final result is that the performance of the model on the standard training set is also reduced. Due to the increase of a large amount of irregular perturbed data, it has a certain effect on the recognition of adversarial examples. However, the accuracy on adversarial

examples is not high, which is lower than the accuracy of the standard model on the standard testing set.

Our method solves the deficiency of the above two methods to a certain extent. SEM shows that the method of uniquely encoding synonyms will reduce the accuracy of the model in the standard testing set, so we adopt the idea of data expansion similar to RSE method and Adversarial Stability Training [13]. Because RSE completely random replacement can not effectively learn the features of the adversarial examples, we adopt the idea of adversarial training to add 10% of adversarial examples in the training set. At the same time, the training set is replaced according to a certain proportion during training, and words are replaced according to a certain proportion in the text selected for perturbation. A small amount of perturbed texts can ensure the performance of the model on the standard dataset and have a better defense effect in adversarial examples. The following is a detailed description of RAT defense methods.

B. Robust Adversarial Training

Our defense framework consists of two steps:

Step 1: Add adversarial examples to the training set

Adversarial training [1, 12] is an effective method to defend against adversarial attacks. In the work of defending against adversarial attacks, researchers usually use adversarial training as a control method [3]. Adversarial training is widely used in the image field [14]. The common method of image adversarial training is perturbing pixels based on gradients during the training process. These perturbations can predict some characteristics of future adversarial examples to achieve the role of defense against adversarial attacks. Due to the difference between text and image, and the main purpose of this paper is defending against the PWWS attack, the adversarial training in this paper is implemented by adding adversarial examples to the existing training set. This method is more targeted and thus has better defense perform on the adversarial examples. Specifically, we first use the PWWS attack to generate k adversarial examples in the standard training set, and then add the adversarial examples to the standard training set, then we get a new training set with enhanced data. Since the new training set contains some adversarial examples from the standard training set, the retrained model can identify some adversarial examples generated by the standard testing set. The value of k will affect the accuracy of the adversarial training model on the standard testing set and the adversarial testing set. We will further discuss the value of k in the subsequent experimental part.

Step 2: Perturb words in training

Although the accuracy of the model for classifying the adversarial examples contained in the training set reaches nearly 100%, it does not have a very good effect in experiments using adversarial examples generated from the testing set, which is also the biggest limitation of adversarial training. In order to learn more about the features of adversarial examples and overcome the randomness of the adversarial example in word replacement, we generate perturbations during the training process by randomly replacing words in the original text. Then we can obtain the perturbed text. In addition, we only perturb part of the text in the training set. We use a small part of the perturbations to ensure that the model can learn enough data features of the standard training set. At the same time, a small part of the perturbed texts and part of the generated adversarial examples can make the model learn a large number features of adversarial examples. These features can guarantee the performance of the model on the adversarial testing set. we use $\beta(s)$ to represent the perturbed text of the original text s, and use $\pi(w_i)$ represent the word after a certain word w_i is replaced:

$$\beta(s) = [\pi(w_1), ..., w_i, ...\pi(w_j), ...\pi(w_n))] \tag{4}$$

We use the method of selecting replacement words in the PWWS attack method to calculate the synonym. Strictly speaking, Our defense method can be regarded as a white-box defense method. The number of perturbed texts generated in the dataset and the number of words replaced by each perturbed text are hyper-parameters, their value will affect the defense results. The influence of these two hyper-parameters on the defense effect will be discussed in detail in the following experimental part. Finally, we use the loss function $L(s, y)$ to promote the classifier to predict the correct label y given the text s. The loss L is a binary or categorical cross-entropy loss with softmax activation:

$$L(s, y) = \sum_{i=1}^{m} -log(y_i|s_i) \tag{5}$$

The entire process of robust adversarial training is shown in Figure 3.

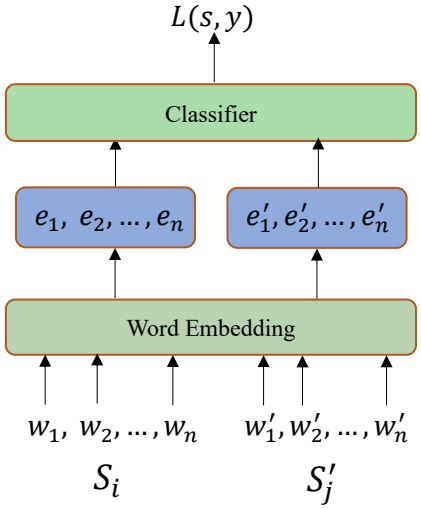


Fig. 3: Robust adversarial training combining original text and perturbed text.

IV. Experiments

A. Experiment setup

1) Experiment Requirements: We use the gpu version of the pytorch deep learning framework in the entire system,

TABLE II: The dataset of our experiments

Dataset	Category	Training set	Testing set	RAT Training set	RAT Testing set	NLP task
IMDB	2	25,000	25,000	27,500	1,000	Sentiment classification
AG's News	4	120,000	7,600	132,000	1,000	Text classification

the version number is 1.2.0. The cuda version is 10.0. The programming language of the entire project is python 3.7. The experimental machine is a personal desktop computer. The GPU is NVIDIA GeForce 1060 6GB, and the CPU is i5-8400. The operating system is ubuntu 16.04.

2) Datasets: We test our framework on two benchmark datasets: IMDB, AG's News.

IMDB [15] contains 50,000 IMDB movie reviews, with 25,000 training sets and 25,000 testing sets, which are dedicated to sentiment classification. The comment labels are classified into two categories. They are pos (positive) and neg (negative). The two labels in the training set and testing set each account for half.

AG's News [16] is a news dataset used for text classification. It contains 4 news categories, namely World, Sports, Business and Sci/Tec. Each news category contains 30,000 training examples and 1,900 testing examples.

On the basis of the above datasets, the training set of RAT has added 10% adversarial examples generated from the standard training set. The adversarial examples used for adversarial training are strictly generated within 25% replacement rate to ensure the effectiveness of adversarial examples. At the same time, they are all examples that are correctly classified by the model to eliminate errors caused by the accuracy of the model itself. The 1000 adversarial examples are used to test the defense model are all randomly generated from the adversarial attack on the standard testing set, which makes training and testing strictly irrelevant. The specific information is shown in Table II.

3) Target model: We use two main classic deep neural networks as base models in our RAT framework for text classification task: LSTM, Bi-LSTM.

LSTM has a 100-dimension embedding layer, two LSTM layers where each LSTM cell has 100 hidden units and a fully-connected layer.

Bi-LSTM also has a 100-dimension embedding layer, two bi-directional LSTM layers and a fully-connected layer. Each LSTM cell has 100 hidden units.

4) Adversarial attack methods: We adopt two synonym replacement adversarial attack models to evaluate the effectiveness of defense methods.

Random. Random replacement adopts the method in the work of [6]. This attack method first randomly selects a group of candidate words with synonyms in the original text, then replaces the original words in the candidate words with random synonyms, and continues to perform this type of replacement until the classification output changes. Since the random replacement attack has only random replacement operation, its attack performance is not very good. In the experiment of this paper, its attack success rate is close to 10%.

PWWS. PWWS is described in detail in section II-A, which is a greedy synonym replacement algorithm that considers the word saliency and the classification probability. As the same, it also only uses synonym replacements but not specific named entities replacements. PWWS attack method has a good attack effect. In our experiments, its attack success rate is close to 70%.

5) The baseline of the comparative experiment: We take No Attack(NT), Adversarial Train(AT) and RSE as three baselines.

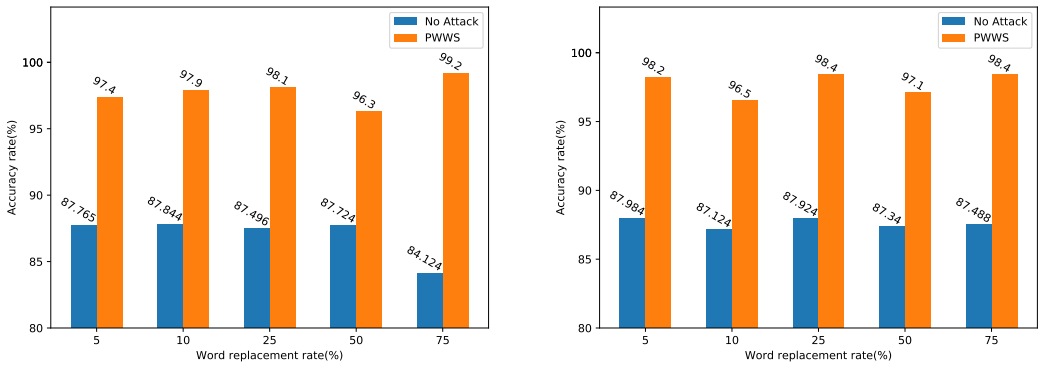
NT is a normal training framework without taking any defense methods.

AT is an adversarial training framework, where extra adversarial examples are generated to train a robust model. We generate 10% adversarial examples from each dataset. and the number of adversarial examples is verified by subsequent experiments. Then the adversarial examples and original training examples are mixed in the training process.

RSE [6] does not change the training set. It makes random replacement of words in the training process to improve the robustness of the model. The detailed description is shown in section II-B recoding, which is also training on the original training set and testing on the original testing set.

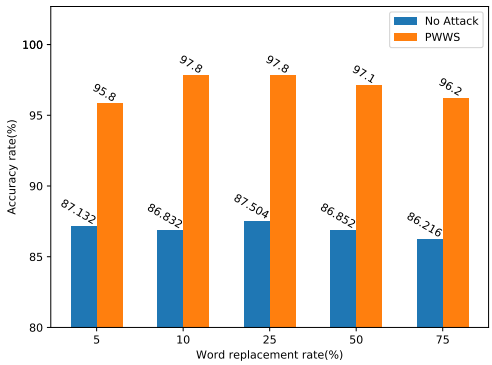
B. Influence of adversarial training with different parameter settings

In this section, we consider the impact of the number of standard training set data expansions on the accuracy of the model in ordinary adversarial training. Compared with the standard LSTM model, we only increase the number of training set, and we temporarily ignore the perturbation operation during the training process in the RAT. We only do experiments on the IMDB dataset and the LSTM model. The experiments are shown in Figure III. Since the generation of adversarial examples is a very time-consuming process, we only generate 15% adversarial examples in the standard training set. It can be seen that the more adversarial examples are added to the training set, the lower the accuracy of the model on the standard testing set. The reason is that the training set added some data with different features from the original data. In addition, the accuracy of the model on 1000 adversarial examples gradually increases, because the more adversarial examples, the better the model learns from adversarial examples. In subsequent experiments, we set the RAT training set to 27500, that is, 10% of the adversarial examples are added. Because this ratio makes model accuracy rarely fall in the standard testing set while the accuracy on the

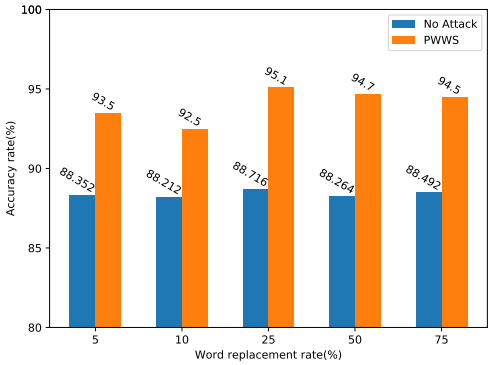


(a) LSTM Accuracy rate when TRR=12.5%

(b) LSTM Accuracy rate when TRR=25%



(c) LSTM Accuracy rate when TRR=50%



(d) Bi-LSTM Accuracy rate when TRR=25%

Fig. 4: Influence of text replacement rate and the word replacement rate.

adversarial example testing set exceeds 90%. The subsequent use of the complete RAT training method can make the accuracy of the model closer to or even exceed the accuracy of the standard model on the standard testing set.

TABLE III: The accuracy of adversarial training

Additional quantity	1,250	2,500	3,750
Standard accuracy(%)	87.024	86.100	85.992
Adversarial accuracy(%)	88.200	91.200	94.100

C. Influence of text replacement rate and the word replacement rate

In section III-B, we mention two factors that affect the defense performance: text replacement rate(TRR) and word replacement rate(WRR). In this section, we verify the effect of the model trained on the IMDB dataset using the RAT method under different TRR and WRR.

Figure 4 shows the experimental results, we can draw the following conclusions:

(1) When the text replacement rate is low, the word replacement rate is large, the accuracy of the model on the standard testing set and the adversarial example testing set is inversely proportional. For example, when TRR=12.5% and WRR=75%, the accuracy of the model on the standard testing set is 84.124%, while the model reaches 99.2% on the adversarial example testing set. This shows that the greater the number of perturbed words, the better the model is learning the features of the adversarial examples, but it loses a lot of information in the standard training set, and finally leads to poor performance on the testing set.

(2)Through experiments on the defense model using different TRR and WRR on the LSTM model, we comprehensively consider the performance of the model on the standard testing set and the adversarial testing set and find that the TRR and WRR set to 25% at the same time is a better parameter setting. When TRR=75%, the accuracy of the model will be further reduced. It is also confirmed on the Bi-LSTM model that a ratio of 25% will make the overall performance of the model better. In subsequent experiments, we set TRR=25% and WRR=25% as the standard settings.

D. The evaluation results on testing set

Table IV shows the performance of the models trained by various defense methods on the standard testing set. Our method has shown excellent performance on both neural networks. It comes close to or even exceeds the original neural network in classification accuracy. Our method outperforms adversarial training and RSE defense in all situations.

TABLE IV: The evaluation results on testing set.

DNNs	Defense	IMDB	AG's news
LSTM	NT	87.616	**89.513**
	AT	86.100	89.118
	RSE	87.084	89.250
	RAT(ours)	**87.924**	89.408
Bi-LSTM	NT	**88.836**	90.263
	AT	88.228	90.566
	RSE	88.140	89.974
	RAT(ours)	88.716	**90.632**

In addition, we show the performance of the method on the standard testing set by using and not using the defense method in the training process. Figure 5 shows the change in accuracy of the model on the standard testing set as the training period increases. We record 100 epochs of the entire training process to observe the situation during the training process. The green curve represents the changes of the standard LSTM model, the red represents the changes of the RAT defense model, and the other two are the adversarial training and the RSE defense model. It can be seen from the figure that the accuracy of all models tends to be stable after the 50th epoch. The RAT model converges faster. The area starts to stabilize at the 40th epoch. In addition, after the 10th epoch, the accuracy of the RAT method in the entire training process is higher than other models because of the data expansion of RAT.

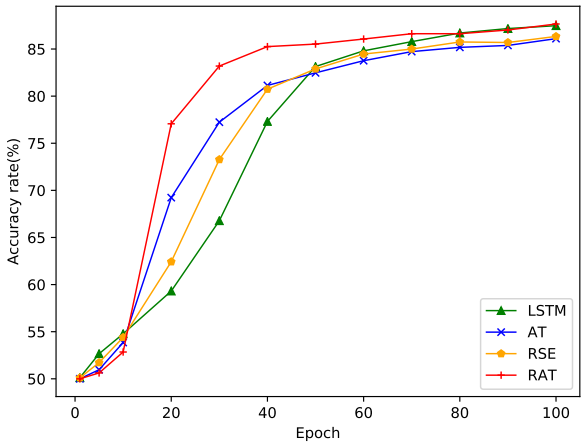


Fig. 5: The change process of accuracy of the four models during training.

E. Defense performance

In this part, we verify the performance of the defense model on the testing set of adversarial examples. We randomly extract data from the standard testing set to form 1000 clean examples for each dataset. Then, these examples are used to generate adversarial examples through the above attack model. These adversarial examples are used as test objects for NT, AT, RSE, and RAT. The evaluation method in this section is still the classification accuracy of the model. The better the effect of the defense model, the higher the accuracy of the test.

Table V shows the accuracy results of different settings, it includes two neural networks: LSTM and BiLSTM; two types of datasets: IMDB and AG's news; two attack methods: random and PWWS; multiple defense methods: AT, RSE, and RAT. The two rows of each dataset are the test results of the defense model on 1000 adversarial examples in the adversarial testing set under the two attack methods. These 1000 adversarial testing sets are all generated from the standard testing set. They are all adversarial examples with correct model classification and replacement rates of less than 25%. From the table, we can draw the following conclusions:

(1) AT, RSE, and RAT have effectively improved the accuracy of the model on the testing set of 1000 adversarial examples, and the accuracy of the original model on the adversarial examples has been increased from 0 to more than 50%, indicating these defense methods All have a certain effect.

(2) Our defense method outperforms existing defense methods. RAT has excellent performance in adversarial examples, and its performance is better than AT and RSE defense methods in two datasets and two attack methods.

V. Conclusion and Future Work

We study the defense work of adversarial examples and investigate the current research status of word-level text adversarial examples in this paper. We find that word-level adversarial examples are more challenging than character-level attacks. The defense method in this paper combined with the idea of adversarial training and perturbing text during training. Its defense effect against adversarial examples is better than the existing word-level defense methods. At the same time, the model generated by the defense method in this paper has only a slight performance fallen and even better than the standard model under some conditions on the testing set. Compared with the existing method, our method is closest to the accuracy of the original model on the testing set.

On the adversarial examples generated by PWWS, the accuracy rate of model recognition is more than 90% on two datasets and two neural networks, which shows that the performance of the model generated by RAT in this paper on the adversarial testing set exceeds the accuracy rate of the standard model on the testing set. The model in this paper is more targeted to adversarial examples.

In addition, the defense method in this paper has only done some work on model enhancement. In the future, we consider combining adversarial example preprocessing to further identify more adversarial examples. The replacement words of adversarial examples will have some unique features, such as in sentiment classification replacement words are often words with obvious emotional tilt. Before the text input into

TABLE V: The evaluation results of 1000 adversarial examples under different settings.

Dataset	Attack	LSTM(%)				BiLSTM(%)			
		NT	AT	RSE	RAT	NT	AT	RSE	RAT
IMDB	Random	0	88.0	77.0	**95.9**	0	72.0	54.0	**75.4**
	PWWS	0	91.2	81.7	**98.4**	0	91.6	62.7	**95.1**
AG's news	Random	0	59.4	57.0	**65.0**	0	67.8	56.6	**69.2**
	PWWS	0	88.3	71.2	**90.0**	0	88.3	64.5	**92.3**

the model, we can check whether the text is perturbed and restore the perturbed words according to a certain algorithm. For example, the pre-trained language model ELECTRA [17] has a replaced token detection mechanism. Effective use of this type of language model will effectively identify the replaced words in the adversarial example. Finally, we can combine replacement word detection and enhanced defense model to further improve the defense success rate, which will be the future work.

REFERENCES

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. B. Estrach, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[2] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 50–56.

[3] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2890–2896.

[4] M. Behjati, S.-M. Moosavi-Dezfooli, M. S. Baghshah, and P. Frossard, "Universal adversarial attacks on text classifiers," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7345–7349.

[5] X. Wang, H. Jin, and K. He, "Natural language adversarial attacks and defenses in word level," *arXiv preprint arXiv:1909.06723*, 2019.

[6] Z. Wang and H. Wang, "Defense of word-level adversarial attacks via random substitution encoding," *arXiv preprint arXiv:2005.00446*, 2020.

[7] J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen, and G. E. Dahl, "Motivating the rules of the game for adversarial example research," *arXiv preprint arXiv:1807.06732*, 2018.

[8] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[9] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2021–2031.

[10] S. Samanta and S. Mehta, "Towards crafting text adversarial samples," *arXiv preprint arXiv:1707.02812*, 2017.

[11] S. Ren, Y. Deng, K. He, and W. Che, "Generating natural language adversarial examples through probability weighted word saliency," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 1085–1097.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[13] H. Liu, Y. Zhang, Y. Wang, Z. Lin, and Y. Chen, "Joint character-level word embedding and adversarial stability training to defend adversarial text." in *AAAI*, 2020, pp. 8384–8391.

[14] U. Shaham, Y. Yamada, and S. Negahban, "Understanding adversarial training: Increasing local stability of supervised models through robust optimization," *Neurocomputing*, vol. 307, pp. 195–204, 2018.

[15] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 142–150.

[16] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[17] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," in *International Conference on Learning Representations*, 2019.