TextShield: A Word Importance-based Ensemble Defense against Word-level Text Attack

Lingfeng Shen*
China Telecom Research Institute
shenlingfengcau@outlook.com

JinPeng Chen†
China Telecom Research Institute
jpchen@bupt.edu.cn

Haiyun Jiang‡
Tencent AI Lab
haiyunjiang@tencent.com

Jie Hu§
China Telecom Research Institute
jiehu1@chinatelecom.cn

Chen Ying¶
China Agricultural University
chenying@cau.edu.cn

ABSTRACT

In natural language processing, deep learning-based models have achieved significant improvements over the last decade. However, neural models are vulnerable to different kinds of adversarial attacks, leading to severe criticism about their reliability. In this paper, we consider the task of text classification and propose TextShield, a defense against word-level text attacks by utilizing the information of word importance. Overall, TextShield is composed of several detectors and one corrector. The detectors aim at judging whether the input sentence is adversarial and the corrector rectifies the adversarial sentences to benign ones, thus guaranteeing the higher performance of text classification. The detectors are designed using the information of word importance which is calculated through the gradient and word frequency. Further, the multiple detectors are also integrated using the random forest to form a more robust detector for the adversarial judgment of an input sentence. Comprehensive experiments show that using TextShield as the defense of adversarial attacks, the neural classification models (e.g., TextCNN, LSTM, and BERT) consistently achieve 8.0%, 4.8%, and 2.8% higher robust accuracy than state-of-the-art defense methods.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Neural networks**.

KEYWORDS

Adversarial Defense, Natural Language Processing

ACM Reference Format:
Lingfeng Shen, JinPeng Chen, Haiyun Jiang, Jie Hu, and Chen Ying. 2018. TextShield: A Word Importance-based Ensemble Defense against Word-level Text Attack. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Deep Neural Networks (DNNs) have obtained great progress in versatile Natural Language Processing (NLP) tasks. Nevertheless, researchers have observed that DNNs are vulnerable to adversarial examples, in which the clean data are modified imperceptibly to humans but could mislead DNN-based models to output wrong predictions [10, 11, 22, 25]. In real-world scenarios where web contents increasingly proliferates, adversary examples often lead to security and safety concerns, so research on defense algorithms is urgently needed.

Text classification is a mainstream task for web-data analysis and also for the robustness check of adversarial attacks; thus, establishing defense methods against adversarial attacks in text classification is valuable and crucial. Furthermore, the most common and effective attack for text classification is word-level attack [9, 20, 40, 45], which is usually implemented by adding, deleting, or substituting words within a sentence. Such an attack often brings catastrophic performance degradation to DNN-based text classification models. Therefore, we choose to study defending against word-level attacks in this paper.

In general, current defense methods can be categorized into three paradigms: (1) model enhancement-based [7, 8, 15, 39], (2) certified robustness-based [14, 18, 44], and (3) detection-based [24] defense.

- Model enhancement-based defense [7, 15, 39] mainly focuses on adversarial training by further retraining the target model with adversarial examples. To guarantee the effectiveness of defense, this paradigm relies on the diversity of additional training data.
- Certified robustness-based defense [14, 18, 44] calculates the boundary of word-level attack, and if a model fits in the boundary, it is robust no matter how adversarial examples are created. However, the calculation of certified robustness is largely affected by models and testing data.
- Detection-based defense [24] often uses a detection-correction pipeline. The detector tries to judge whether the sentence is adversarial, and the corrector corrects the sentence if it is adversarial. In [24], the detector is built mainly by using the frequency difference between the original word and the adversarial word (the word used to substitute the original word).

In this paper, we design the defense method under the detection-correction pipeline. The key research issue of detection-based defense is how to develop effective detectors. Though word frequency [24] has shown its power in adversarial detection, it will be much less

effective in the case that the frequency difference between the original word and the adversarial word is not distinctive. Thus frequency-based detector is extremely insufficient. In this work, we propose TextShield, a detection-based defense to defend against word-level attacks in text classification.

In order to tackle the frequency difference problem in detectors in [24], we follow the progress in the deep learning explainability field [2, 37, 38] and use the word-level information that well reflects the word importance to develop detectors. For an input sentence, we investigate the impact of each word on determining the label of the sentence, namely word importance. However, existing methods for word importance are model-agnostic, task-agnostic (e.g., 'excellent' is vital in sentiment classification but is unimportant in category classification.), and sentence-agnostic (e.g., the important word in one sentence may be trivial in other sentences.), such properties fail to comprehensively reflect the importance of one word under a specific setting. Therefore, we propose a model-specific, task-specific, and sentence-specific method which utilizes the gradient information in DNN-based models to better measure the word importance. Moreover, since vanilla gradient (VG) may be insufficient to reflect word importance [31, 32], three improved gradient computation methods are also additionally used to calculate word importance, respectively: guided backpropagation (GBP) [37], layerwise relevance propagation (LRP) [23] and integration gradient (IG) [38]. In our framework, we develop four gradient-based detectors using corresponding gradient method.

Furthermore, to strengthen the performance, an ensemble detector is built, which combines the four gradient-based detectors and a frequency-based detector (based on the frequency of adversarial words) by a random forest mechanism, thus yielding a more robust ensemble detector. The ensemble detector gives a final judge of whether an input sentence is adversarial. Then, the corrector rectifies the candidate words with a high adversarial probability in the sentence into their high-frequency synonyms. Instead of the original sentence, the sentence after correction will be fed to the text classifier.

TextShield owns two principal advantages: (1) TextShield does not hold any assumptions for word-level text attack, thus free of the prior knowledge of the coming text attack. (2) TextShield utilizes the specifically designed word importance, which can be well adaptive to different settings (e.g., model), thus possesses better defense capacity. Our experiments on three popular text classification benchmarks show that TextShield outperforms the state-of-the-art (SoTA) defense method and it is a model-agnostic defense method.

The main contribution of this paper are summarized as follows:

- We design TextShield, a detection-based defense against word-level attacks, where five detectors are designed, and a random forest is used to combine them for adversarial sentence detection. Besides, a corrector is also built to rectify adversarial sentences into benign ones.
- We propose word importance to connect the adversarial robustness and the gradient information of DNN-based models and then leverage such a connection to design four gradient-based detectors. Such a design makes TextShield free of prior knowledge about the text attack and remains effective when the frequency change is not distinctive.
- Our proposed TextShield demonstrates superior performance than SoTA defense methods under various neural models on three text classification benchmarks.

Throughout the rest of the paper, we firstly discuss the related works in Sec 2 and the relations between TextShield and them. Then we illustrate our method in Sec 3. Specifically, we present an overview in Sec 3.1, and demonstrate the details of gradient-based detector, frequency-based detector, ensemble mechanism, and corrector in Sec 3.3, Sec 3.4, Sec 3.5 and Sec 3.6, respectively. To evaluate the effectiveness, we show and discuss the experimental results in Sec 4. Finally, we discuss the effectiveness of the components of TextShield in Sec 5.

2 RELATED WORK

2.1 Defense against Word-level Text Attack

Model enhancement is a popular way to improve the model's robustness against text attacks, and most of them focus on adversarial training methods which re-train a victim model with additional adversarial examples to obtain an enhanced model. Alzantot [1], and Ren [31] generated adversarial examples by an attack method and re-trained models which are more robust against the attack. Unfortunately, the robustness of re-trained models is often determined by the diversity of adversarial examples used in re-training. As a result, adversarial training is vulnerable to unknown attacks, which is unrealistic in real-world scenarios.

Besides, a stream of recent popular defense methods [13, 14] concentrate on certified robustness, which ensures models are provably robust to all potential word perturbations within the constraints. For example, IBP [14] performs catastrophically on large pre-trained language models (e.g., BERT). However, because of the extreme time cost in the training stage, certified robustness is difficult to be applied to complex models [14].

Recently, some work [24] tries to design detection-based defense methods . Mozes [24] used the frequency difference between the original word and the adversarial word to design detectors. Despite their success, this method is less effective when the frequency difference between the words is not distinctive.

2.2 Word Importance

The research of measuring word importance can be dated back to tf-idf [16, 21], and weights based on variants of word occurrence frequency or tf-idf have been well studied in literature [29, 34, 35]. However, these methods have the following three limitations which lead to less effectiveness in describing word importance: not model-specific, not task-specific, and not sentence-specific. In this paper, we use gradient information to calculate word importance to strengthen the detectors since there is a cross-cutting link between gradient and adversarial robustness.

3 METHODOLOGY

3.1 Preliminaries

In this section, we present some basic definitions used in our framework. In the text classification scenario, $X = \{w_1, w_2, \ldots, w_d\}$ is a sentence with d words, and $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d$ are the corresponding word embeddings. X is labeled with a class $y \in \mathcal{Y}$ with $|\mathcal{Y}| = N$.

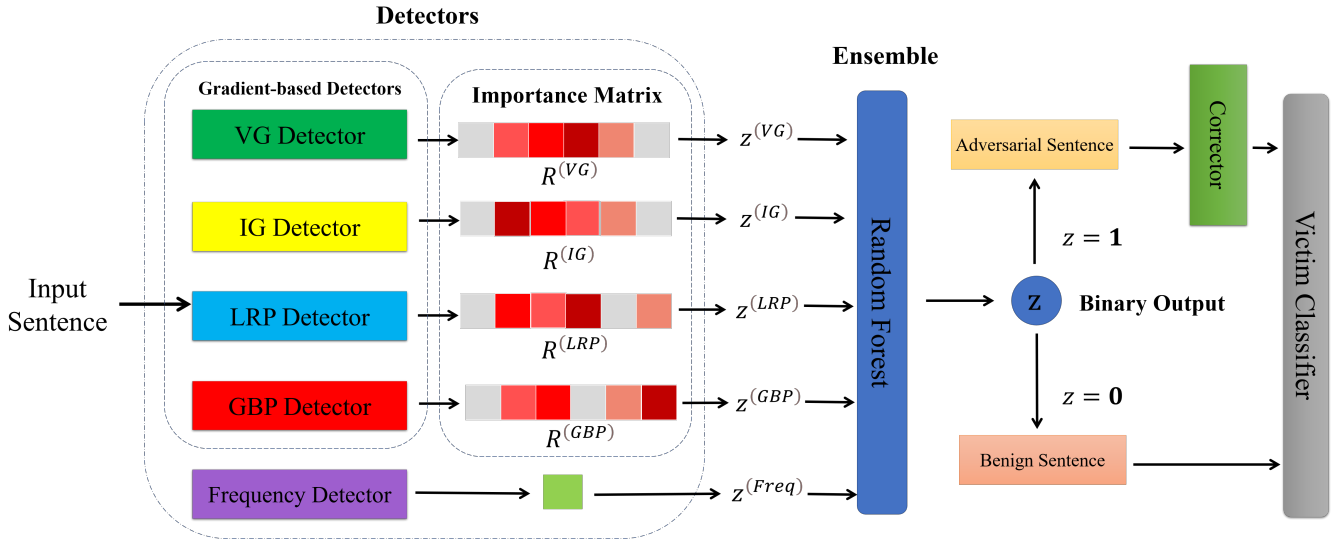


Figure 1: The overview of TextShield. In the importance matrix, the darker color means the word is more important to the prediction. VG detector means that this detector calculates word importance through vanilla gradient (VG).

Given a text classifier F, $F(X)$ denotes the probability distribution over the class set $\mathcal{Y}$ for the input sentence X. We select the class y_j with the largest probability in $F(X)$ as the predicted label of X.

Generally, in a common text attack setting, there is a *victim classifier* and a *text attack*.

- Victim Classifier: Given a set of training data in text classification, a classifier $F(x)$ can be easily trained, where the sentence encoder is usually embodied as CNN [17], LSTM [12] or BERT [6].
- Benign Sentence: The original sentences in a text classification benchmark.
- Text Attack: For each benign sentence X with a true label y_j, the text attack $G(\cdot)$ will generate an adversarial sentence $X^* = G(X)$ by substituting words w_i in X with w_i^*, satisfying: (1) $F(X^*)$ and $F(X)$ output different class labels with high probability and (2) $Dist\left(\mathbf{w}_i, \mathbf{w}_i^*\right) < \delta$. $Dist(\cdot,\cdot)$ is a distance metric and δ is a threshold to limit the distance between two words. For example, $Dist(,)$ can be the l_2 distance between two words' embedding. Intuitively, a small distance indicates that the two words are likely to be synonyms.
- Adversarial Sentences: The sentences that generated by a text attack, which aims to mislead the victim classifier.

3.2 Overview

Generally, for an input sentence X, TextShield takes an ensemble detector (consists of one frequency-based detector and four gradient-based detectors) to judge whether it is benign or adversarial. If X is benign, it will be fed to the victim classifier F directly. Otherwise, TextShield uses the corrector to recover X to $X^{(c)}$ and then feed $X^{(c)}$ to F for prediction. The overview framework is shown in Figure 1, *which is like a shield in front of the victim classifier.* Generally, TextShield has two principal components: Ensemble Detector and Corrector.

3.2.1 Ensemble Detector. The word importance information plays an very important role in our framework, and the ensemble detector is integrated by two kinds of detectors: four gradient-based detectors and one frequency-based detector. In our framework, we take the random forest algorithm [5] to integrate these detectors. Specifically, we use a random forest (RF) to combine different detectors. Note that the ensemble is novel for combining different detectors for robustness enhancement.

Gradient-based detector. For a sentence X with d words, a gradient-based detector first produce a word importance matrix $\mathcal{R} = \mathcal{I}(X) \in \mathbb{R}^{d \times N}$, where $\mathcal{I}$ represents the method for word importance calculation, and the (i, j)-th element in $\mathcal{R}$ (denoted as R_{ij}) represents the importance of the i-th word towards the j-th calss label. That is, r_{ij} reflects the importance of word w_i for F if predicting sentence X as y_j. Then, the detector takes the importance matrix $\mathcal{R}$ as input and generates an output z.

$$z = D(\mathcal{R}) \tag{1}$$

where D is an LSTM encoder and the details will be presented in the following sections. $z < 0.5$ indicates that X is a benign sentence and $z > 0.5$ means an adversarial one.

Instead of using only one solution to compute the word importance matrix, we obtain the importance matrix using different gradient calculation strategies, i.e., VG, GBP [37], LRP [2] and IG [38].

Frequency-based detector. Since gradient information is not explicitly relevant to the linguistic property of texts, we also incorporate a frequency-based detector to enhance TextShield. As a result, the frequency information can well complement gradient information for adversarial detection.

For a sentence X with prediction label y, the frequency-based detector first generates a new sentence X° by replacing the rare words in X with their synonyms that own higher frequencies in the

training corpus. Then it checks whether $|F_y(X) - F_y(X^\circ)| > \alpha$, where $F_y(X)$ is the probability of class y predicted by F for the sentence X. α is a predefined threshold. If the inequality holds, the frequency-based detector will recognize it as an adversarial sentence.

3.2.2 Corrector. If an input sentence X is predicted as adversarial by the ensemble detector, the corrector will generate a new benign sentence X' based on X by substituting specific words with their frequent synonyms. Here, the specific words refer to the ones with high word importance. Then, the new sentence X' will be fed into the classifier for prediction.

3.3 Gradient-based Detectors

3.3.1 Intuition. Intuitively, given a sentence, adding the perturbation to the most important word is more effective than adding to the unimportant one [31]. Therefore, focusing on such important words might be effective for adversarial detection. We design a toy experiment. We use three text attacks: GSA [19], PWWS [31], GA [1] to generate adversarial sentences for a victim classifier, the details of the toy experiment are deferred to the appendix. Specifically, we select the adversarial sentences that successfully mislead the victim classifier and mask the word that owns largest $|r_{ij}|$, where $|r_{ij}|$ is the importance w.r.t the prediction $y_j \in [N]$, and calculation of $|r_{ij}|$ is illustrated as follows:

Table 1: Accuracy after masking 10% most important words. 'Original' represents original accuracy while 'Modified' represents accuracy after masking.

Accuracy	GA	PWWS	GSA	Avg
Original	0	0	0	0
Modified	31.7	30.3	29.8	30.6

$$|r_{ij}| = |\frac{\partial F_{y_j}(X)}{\partial w_i}|$$

where F_{y_j} is the j -th element of confidence. A larger value of $|r_{ij}|$ indicates that the word w_i is a more important word for classifying the sentence X as the class y_j.

As shown in Table 1, model's accuracy on such adversarial sentences is 0% while the modified accuracy increases to 30.6%. This toy experiment confirms our intuition that $|r_{ij}|$ can be used to detect adversarial sentences.

The main reason why gradient information is highly related to adversarial robustness is that the gradient $|r_{ij}|$ plays a crucial role in adversarial attack process. For a benign sentence X and a victim classifier F, the objective of a text attack which leads to a wrong prediction $y^{(j)}$ is defined as follows:

$$\arg\min \mathcal{L}\left(F_{y_j}\left(X^*\right), y_j\right) \tag{2}$$

where $\mathcal{L}$ is a loss function (e.g., cross-entropy loss).

In ease of clarification, we regard the word embeddings as continuous variables. In a vanilla gradient-based attack [10], Eq. (2) can be optimized by a gradient descent method, the word embedding $\mathbf{w_i}$ is perturbed with a step rate r_1 as follows:

$$w_i^{'} = w_i - r_1 \frac{\partial \mathcal{L}(F_{y_j}(X), y^{(j)})}{\partial w_i} \tag{3}$$

After an iteration, the attack substitutes the word w_i with the word whose embedding is closest to $\mathbf{w_i^{'}}$, such an operation can probably fool the classifier. Then, we further derive the term $w_i - r_1 \frac{\partial \mathcal{L}(F_{y_j}(X), y^{(j)})}{\partial w_i}$, and we rewrite it as follows:

$$w_i - r_1 \frac{\partial \mathcal{L}(F_{y_j}(X), y^{(j)})}{\partial F_{y_j}(X)} \cdot |r_{ij}| \tag{4}$$

As shown in Eq. (4), words with larger $|r_{ij}|$ are perturbed more severely by text attacks. Motivated by such a connection between attacks and the gradient information of DNNs, we see that the importance $|r_{ij}|$ can be helpful for adversarial detection. In one word, $|r_{ij}|$ may serve as keys for adversarial sentence detection.

3.3.2 Design of Gradient-based Detectors. Based on the intuition, we design detectors that utilize gradient information. As introduced in Section 3.2, the detector firstly outputs an importance matrix by a computation method $\mathcal{I}$, and then produces a signal z to indicate whether it is an adversarial sentence.

Specifically, we use gradient information to calculate word importance $|R_{ij}|$, and there are four methods: vanilla gradient (VG), guided backpropagation (GBP) [37], layerwise relevance propagation (LRP) [2], and integration gradient (IG) [38].

Vanilla Gradient (VG). For a text classifier F, the importance of word w_i in X to the prediction y_j by the vanilla gradient is calculated as follows:

$$\mathcal{R}_{ij}^{(VG)} = \frac{\partial y_j}{\partial w_i} \tag{5}$$

where $\mathcal{R}^{(VG)} \in \mathbb{R}^{d \times N}$ is the importance matrix. The (i, j) -th element of R, indicates the importance of word w_i to the class j. Specifically, when calculating the gradient, y_j is a scalar and $w_i \in \mathbb{R}^{1 \times k}$, where k is dimension of word embedding. In fact, $\frac{\partial y_j}{\partial w_i}$ is a vector with the form $\mathbb{R}^{1 \times k}$, and we make an average operation to obtain $\mathcal{R}_{ij}^{(VG)}$. For brevity, we omit the operation in our paper.

Guided Backpropagation (GBP). Based on the idea of vanilla gradient, GBP [37] only chooses the 'positive' (> 0) gradient information instead of 'negative' (< 0) gradient information. Positive gradients indicate supporting the prediction while negative ones mean the opposite, such an operation enables us to capture what classifier F learns but discards anything that F does not learn.

Specifically, when propagating the gradient, GBP [37] sets all the negative gradients to 0, and finally calculates the gradient of input towards prediction. We take the final outputs of GBP [37] as the importance matrix $\mathcal{R}^{GBP}$. Details of GBP can refer to [37].

Layerwise Relevance Propagation (LRP). GBP owns obvious drawbacks called gradient saturation [2, 4] that the gradients computed by GBP are always zeros under some cases. Therefore, Layerwise Relevance Propagation (LRP) [2] is proposed.

LRP points out this problem and highlights the importance of having a reference input besides the target input to calculate the gradient. The LRP method calculates a weight for each connection in the gradient chain. It uses a reference input instead of roughly setting

the weights as 0 or 1 like GBP. The reference input is a 'neutral' input and will be different in different tasks (e.g., blank images in CV field or zero embedding in NLP field). Specifically, it aims at decomposing classifier's confidence to calculate each variable's (e.g., word, pixel) contribution since it regards the confidence of a sum of the gradient of each input variable. Finally, we take such gradients as $\mathcal{R}_{ij}^{LRP}$. Details can refer to [37].

Integrated Gradient (IG). Some works [38] pointed out that LRP is calculating the discrete gradient. However, the chain rule does not hold for discrete gradients thus LRP has essential drawbacks. The integrated gradient (IG) [38] calculates the influence of word w_i to prediction y_j by integration of vanilla gradient, deined as follows:

$$\mathcal{R}_{ij}^{(IG)} = \left(\mathbf{w_i} - \mathbf{w_i}'\right) \times \int_0^1 \frac{\partial F_{y_j}\left(X, \mathbf{w_i}' + \alpha\left(\mathbf{w_i} - \mathbf{w_i}'\right)\right)}{\partial \mathbf{w_i}} \, \mathrm{d}\alpha \quad (6)$$

where w_i' is a zero embedding and $F_{y_j}(X, \mathbf{w_i}' + \alpha(\mathbf{w_i} - \mathbf{w_i}')$ means replace w_i's embedding $\mathbf{w_i}$ with $\mathbf{w_i}' + \alpha(\mathbf{w_i} - \mathbf{w_i}')$.

Training of gradient-based detector. We train an LSTM to classify sentence X as a benign or adversarial sentence for each detector. The LSTM's input is importance matrices $\mathcal{R}_{\cdot j}$ generated by the different methods (i.e., VG), which is defined as follows:

$$\begin{aligned} z^{(VG)} &= \mathrm{LSTM}_1\left(\mathcal{R}_{\cdot j}^{(VG)}\right), \quad & z^{(IG)} &= \mathrm{LSTM}_2\left(\mathcal{R}_{\cdot j}^{(IG)}\right) \\ z^{(GBP)} &= \mathrm{LSTM}_3\left(\mathcal{R}_{\cdot j}^{(GBP)}\right), \quad & z^{(LRP)} &= \mathrm{LSTM}_4\left(\mathcal{R}_{\cdot j}^{(LRP)}\right) \end{aligned} \quad (7)$$

where z is confidence to judge whether the sentence is adversarial.

3.4 Frequency-base Detectors

3.4.1 Intuition. Given a sentence, substituting the target word with its rare synonym rather than common synonym increases attack effectiveness. For example, on the IMDB dataset [28], attack GA substitutes the word 'happy' with 'hilarious' to generate a successful adversarial sentence, but substituting 'happy' with 'glad' fails to mislead the classifier. The key is that the frequency difference between 'glad' and 'hilarious' is large in the train set.

Table 2: Accuracy after substituting rare words with more frequent words. 'Original' represents original success rate while 'Modified' represents accuracy after substitution.

Accuracy	GA	PWWS	GSA	Avg
Original	0	0	0	0
Modified	19.9	29.3	21.8	23.7

Also, we design a toy experiment to validate our intuition. We select adversarial sentences that successfully attack the victim model by GSA, PWWS, and GA. The details of the toy experiment are also deferred to the appendix. By substituting the words with synonyms with higher frequencies in the training corpus, as shown in Table 2, the accuracy rises to 23.7%, validating our intuition.

3.4.2 Design of Frequency-base Detector. Inspired by the intuition, our frequency-based detector converts a sentence X into X° by replacing random words with their frequent synonyms.

Firstly, we define a function $\phi(w_i, w_j) = \frac{p(w_i)}{p(w_j)}$ that indicates the frequency difference between two words w_i, w_j, $p(w_i)$ means the frequency of w_i in the training corpus. Then we denote that w_j is w_i's synonym if their word embeddings satisfy $||\mathbf{w_i} - \mathbf{w_j}||^2 \leq \delta$, where δ is a hyper-parameter, and we let $\mathcal{S}(w_i) = \{w : ||\mathbf{w_i} - \mathbf{w}||^2 \leq \delta\}$ denote the synonym set for w_i.

Then frequency-based detector generates a sentence X° from X by substituting random 10% words w_i with more frequent words from $\mathcal{S}(w_i)$. Specifically, for each word $w \in \mathcal{S}(w_i)$ we select word w with highest frequency: $w = \mathrm{argmin}_{w \in \mathcal{S}(w_i)} \phi(w_i, w)$ and substitute w_i with w. Therefore, X° is obtained after the substitution.

Finally, the frequency-based detector outputs a value z^{freq} to indicate whether the sentence is adversarial. For a label y_j of X and a threshold $\alpha \in [0, 1]$, the sentence X is considered adversarial if $F(X)_{y_j} - F(X^\circ)_{y_j} > \alpha$. The sensitivity towards threshold α is recognized as sensitivity to word frequency [24], which will be discussed in the ablation study.

3.5 Ensemble Mechanism

After designing four gradient-based detectors and one frequency-based detector, we use a random forest to integrate the five detectors as an ensemble detector, which is defined as follows:

$$z = RF\left(z^{(freq)}, z^{(VG)}, z^{(IG)}, z^{(GBP)}, z^{(LRP)}\right) \quad (8)$$

The output z is the final judgment for adversarial detection; if $z < 0.5$, the sentence will be recognized as benign, and $z > 0.5$ means the opposite. With such an ensemble mechanism, the ensemble detector is more robust than a single detector when facing various kinds of text attacks.

3.6 Corrector

If a sentence is an adversarial one, the sentence will be fed to the corrector. As discussed above, given an importance matrix, larger $\mathcal{R}_{ij}$ means the word w_i is more important for classifying the sentence as y_j. Specifically, $\mathcal{R}_{ij}$ is $\mathcal{R}_{ij}^{VG}$ in the corrector. If the ensemble detector recognizes the input sentence as an adversarial one, then the prediction class j is possibly inaccurate, so the words with large $\mathcal{R}_{ij}$ are possible to be perturbed by attacks to fool the classifier. Therefore, the corrector corrects the words with a large $\mathcal{R}_{ij}$ in the adversarial sentence. Specifically, for an adversarial sentence X^* with its prediction y_j, the corrector regards a word as a suspect when its $\mathcal{R}_{ij}$ exceeds a threshold, which is defined as follows:

$$\mathcal{R}_{ij} > \beta \cdot (\mathcal{R}_{\max} - \mathcal{R}_{\min}) + \mathcal{R}_{\min} \quad (9)$$

where $\mathcal{R}_{\max}$ and $\mathcal{R}_{\min}$ are the largest and smallest values in $\mathcal{R}$, and β is a hyper-parameter to control the ratio of suspect. For a suspect w_i, the corrector substitute w_i with the most frequent word $w' \in \mathcal{S}(w_i)$. Then the sentence is fed to the victim classifier.

3.7 Training of TextShield

For a benign sentence set $\mathcal{S}_x = \{X_1, \ldots, X_N\}$, we use adversarial attacks to generate and select adversarial sentences that successfully fools the classifier. The training of TextShield which is a binary classification task (adversarial sentences vs. benign sentences) is composed of two stages. The first stage is to train the LSTMs for

Table 3: The accuracy (%) of various models on three benchmarks, on benign sentences and adversarial sentences generated by different attacks. The numbers in bold indicate the best performance with the statistical significance threshold of p-value 0.05 in the paired t-test. No: benign sentences, NT: Normal Training, AT: Adversarial Training, TS: TextShield

Dataset	Attack	TextCNN						LSTM						BERT				
		NT	AT	IBP	FGWS	SEM	TS	NT	AT	IBP	FGWS	SEM	TS	NT	AT	FGWS	SEM	TS
IMDB	No	88.7	**89.1**	78.6	84.7	86.8	88.5	87.3	**89.6**	79.5	80.9	86.8	85.5	92.3	**92.5**	88.7	89.5	88.0
	GSA	13.3	16.9	72.5	-	66.4	**75.5**	8.3	21.1	70.0	-	72.2	**75.0**	24.5	34.4	-	89.3	**92.0**
	PWWS	4.4	5.3	72.5	68.5	71.1	**76.5**	2.2	3.6	70.0	57.1	77.0	**81.5**	40.7	52.2	48.9	89.3	**92.5**
	GA	7.1	10.7	71.5	70.2	71.8	**77.5**	2.6	9.0	69.0	54.9	77.0	**82.0**	40.7	57.4	55.0	89.3	**92.0**
	IGA	2.3	-	65.0	-	72.4	**78.0**	2.7	-	64.5	-	73.8	**78.7**	28.3	-	-	85.5	**91.0**
AG's News	No	**92.3**	92.2	89.4	90.2	89.7	90.5	92.6	92.8	86.3	87.5	90.9	91.0	94.6	**94.7**	93.5	94.1	93.6
	GSA	45.5	55.5	86.0	-	80.0	**88.5**	35.0	58.5	79.5	-	85.5	**88.5**	66.5	74.0	-	88.5	**89.0**
	PWWS	37.5	52.0	86.0	76.5	80.5	**89.0**	30.0	56.0	79.5	75.5	86.5	**89.2**	68.0	78.0	72.3	88.5	88.0
	GA	36.0	48.0	85.0	72.5	80.5	**88.5**	29.0	54.0	76.5	80.5	85.0	**87.5**	58.5	71.5	75.0	88.5	**90.5**
	IGA	30.0	-	86.0	-	80.0	**89.0**	26.5	-	79.5	-	85.5	**89.0**	45.5	-	-	88.5	**90.0**
Yahoo! Answers	No	68.4	**69.3**	64.2	65.0	65.8	66.0	71.6	**71.7**	51.2	68.0	69.0	68.5	**77.7**	76.5	75.8	76.2	75.9
	GSA	19.6	20.8	61.0	-	49.4	**64.5**	27.6	30.5	30.0	-	48.6	**64.0**	31.3	41.8	-	66.8	**70.6**
	PWWS	10.3	12.5	61.0	52.5	52.6	**64.0**	21.1	22.9	30.0	50.2	54.9	**64.0**	34.3	47.5	50.6	66.8	**70.0**
	GA	13.7	16.6	61.0	53.0	59.2	**64.5**	15.8	17.9	30.5	45.5	66.2	**67.0**	15.7	33.5	55.5	66.4	**70.0**
	IGA	3.5	-	59.0	-	61.0	**65.5**	5.5	-	31.5	-	62.5	**66.5**	7.0	-	-	62.0	**66.5**

gradient-based detectors, and each LSTM is trained independently. After training the LSTMs, we train the random forest in the second stage with the same setting. The detailed settings are reported Sec 4.4.

4 EXPERIMENTS

4.1 Datasets and Text Attacks

We select three popular benchmarks in text classification, which are also widely used in text attack and defense.

- **AG's News** [48]: AG's News contains news of four topics: World, Sports, Business and Sci/Tech. Each class contains 30000 training samples and 1,900 testing samples.
- **Yahoo! Answers** [48]: Yahoo! Answers consists of texts from 10 topics. Each topic contains 140000 training samples and 5000 testing samples.
- **IMDB** [28]: IMDB is a binary sentiment classification dataset. Each class contains 25000 training samples and 25000 testing samples.

For word-level text attacks, we select four adversarial attacks, which are widely used in the stream of adversarial robustness in the NLP field: GSA [19], PWWS [31], GA [1] and IGA [41]. All the settings of four attacks are the same as their original papers, respectively.

4.2 Baselines

We choose four defense methods as the baselines: (1) adversarial training [10]: the most common defense in the model-enhancement paradigm. (2) IBP [14]: the most representative method in certified robustness paradigm. (3) synonym encoding method (SEM) [42]: the SoTA defense method against word-level attack. (4) frequency-based word substitution (FGWS) [24]: the SoTA method in the detection paradigm.

4.3 Victim Classifier

To validate that our defense method can fit into various types of victim models, we use different DNN-based victim classifiers, including TextCNN [17], LSTM [12], and Bert [6]. For TextCNN and LSTM, we use Glove [27] for the pre-trained embedding with a size of 300. Specifically, TextCNN has three convolutional layers with the filter sizes of 3, 4, and 5, a dropout layer (dropout rate = 0.2), and a fully connected layer. The LSTM-based victim model has two layers, each with 128 cells. Bert is 'Bert-base' with the checkpoint from Huggingface. Moreover, for a fair comparison, as the previous work [36] reported that BERT is too difficult to be tightly verified by IBP, we do not use IBP on BERT.

4.4 Training Settings

To train TextShield, we need to use text attacks to prepare adversarial examples. We use adversarial attacks to generate and select 2000 adversarial sentences that successfully attacks the model, equally generated by GA, IGA, GSA and PWWS, and the attacks are implemented by [46]. The training process of TextShield is detailed in Sec 3.7. As for hyper-parameter settings, β is set to 0.4 and δ is set to 0.5. The LSTM used in the corrector (Eq 9) has two LSTM layers where each layer has 128 LSTM cells and a fully-connected layer. Random forest is implemented by [26], the max depth is set as 3 and other parameters are default, considering that $z^{(freq)}$ is a discrete feature, we use a one-hot encoder [26] to transform.

The training settings of baselines are listed as follows. We select 2000 adversarial sentences that successfully attack the model for adversarial training, which are the same as TextShield training. As for IBP, the settings are kept the same as in its original paper. In FGWS, the threshold is set to 1.0; In SEM, distance δ is set to 0.5, and synonym number is set to 10.

4.5 Defence of TextShield

During evaluation, we sample 1000 benign sentences from each benchmark and use the above text attacks to generate 1000 adversarial sentences[1]. Then we evaluate the performance of victim models with or without defense methods. The more powerful the defense method is, the better classification accuracy on adversarial sentences is. Table 3 illustrates the performances of various defense baselines on benign sentences and adversarial sentences.

When evaluating the benign sentences, adversarial training (AT) can improve the classification accuracy of most models on three

[1]There is no intersection between these 1000 sentences and 2000 sentences in training.

Table 4: The performance (%) of various models for adversarial examples which are generated by a specific model on AG's News to evaluate transferability. * represents that the adversarial sentences are generated by the model in the column. The numbers in bold indicate the best performance with the statistical significance threshold of p-value 0.05 in the paired t-test.

Attack	TextCNN						LSTM						BERT				
	NT	AT	IBP	FGWS	SEM	TS	NT	AT	IBP	FGWS	SEM	TS	NT	AT	FGWS	SEM	TS
GSA	45.5*	86.0	87.0	-	87.0	**88.5**	80.0	89.0	83.0	-	91.0	**92.0**	92.5	**94.5**	-	90.5	91.0
PWWS	37.5*	86.5	87.0	78.5	87.0	**88.5**	70.5	87.5	83.0	80.1	90.5	**92.5**	90.5	**95.0**	86.4	90.5	91.0
GA	36.0*	85.5	87.0	82.5	87.0	**87.5**	75.5	88.0	83.5	82.4	90.5	**90.5**	91.5	**95.0**	83.0	90.5	92.0
IGA	30.0*	82.5	85.0	-	86.5	**87.0**	74.5	86.4	83.5	-	90.5	**92.0**	90.0	**94.0**	-	90.5	91.0
GSA	84.2	89.0	87.5	-	87.0	**88.5**	35.0*	87.0	83.5	-	90.5	**92.5**	93.2	**95.5**	-	90.5	91.7
PWWS	83.1	89.0	87.5	76.5	87.0	**89.0**	30.0*	86.0	83.0	78.5	90.5	**91.2**	92.5	**94.9**	82.9	90.5	92.3
GA	84.4	89.5	87.5	72.5	87.0	**88.5**	29.0*	88.0	83.5	80.5	90.5	**91.7**	92.1	**95.2**	80.1	90.5	92.4
IGA	82.2	88.2	87.5	-	87.0	**88.5**	26.5*	85.0	79.5	-	90.5	**90.9**	91.3	**93.7**	-	90.5	91.8
GSA	83.5	87.0	87.5	-	87.0	**88.5**	84.0	88.0	83.5	-	89.5	**90.7**	66.5*	**95.5**	-	90.5	91.5
PWWS	81.0	87.5	88.0	84.2	87.0	**89.5**	82.5	88.0	84.0	80.8	91.5	**91.5**	68.0*	**94.5**	82.6	90.5	90.5
GA	82.0	87.0	88.0	83.1	87.0	**89.7**	82.0	88.0	83.5	76.8	91.0	**91.7**	58.5*	**94.0**	81.6	90.0	90.5
IGA	80.2	86.7	86.7	-	86.0	**89.7**	81.2	88.0	82.5	-	89.2	**91.5**	58.3*	**94.0**	81.9	89.8	91.0

datasets, which indicates AT can also enhance the generalization of training. IBP achieves much lower accuracy on benign data due to its high complexity and relatively loose bounds. Our proposed method, TextShield, achieves an accuracy close to normal training, with a tiny trade-off between robustness and generalization. Such a trade-off is usual for attack/defense in the CV field, which has been theoretically investigated [47].

Although NT and AT achieved high accuracy on benign sentences, the accuracy of both NT and AT significantly decreased under text attacks on adversarial sentences. For NT, the accuracy drop exceeds 70%, 50%, and 50% on the three datasets. At the same time, AT is extremely powerless to defend against these attacks, especially for PWWS and GA. Specifically, AT can only improve accuracy by $5-10\%$ on adversarial sentences, far below the requirements of modern NLP models for robustness. One possible reason is that AT requires data diversity [31], and limited sentences are unable to meet the requirements.

As for IBP, it should be noted that the IBP-like methods [30, 43] which were firstly proposed in the CV domain is more suitable for CNN and does not perform excellently on LSTM. Moreover, IBP is difficult to train and does not perform perfectly on either benign or adversarial examples. FGWS, on the other hand, focuses only on word frequencies, which is not sufficient to detect adversarial samples well. SEM achieves good results by synonym encoding, and however, TextShield demonstrates stronger defense capacity than SEM in almost every setting. Specifically, TextShield achieves 8.0%, 4.8%, and 2.8% higher accuracy than SEM.

4.6 Transferability of TextShield

Text attacks show transferability across different DNN-based models, in which adversarial examples targeting one model may also mislead other models. The transferability of text attacks matters because it may attack a real-world DNN application without knowing the victim model. Therefore, a reliable defense method should block the transferability of adversarial sentences.

As shown in Table 4, TextShield is more successful in defending the transferability of adversarial sentences than the baselines on TextCNN and LSTM. For Bert, the transferability of adversarial sentences generated on other models performs very weakly, and the accuracy here relies more on the powerful generalization of Bert, so AT achieves the best performance, which is similar to previous works [3, 33, 41]. Generally, TextShield can significantly improve the model robustness and defend the transferability of adversarial sentences successfully.

5 ABLATION STUDY

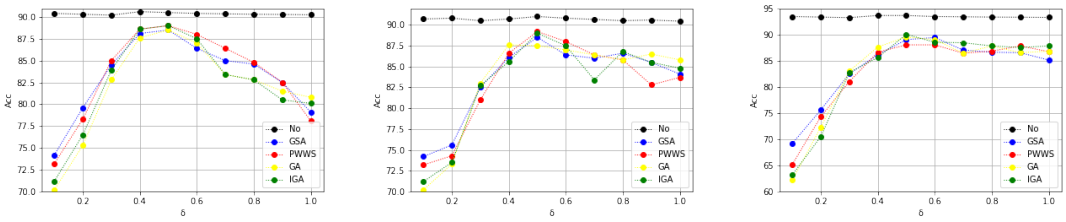
This section illustrates ablation studies of TextShield, consisting of three parts: (1) analysis of hyper-parameters (2) analysis of ensembling (3) analysis of training samples. Specifically, the analysis of training samples is deferred to the appendix due to limited space.

5.1 Analysis of hyper-parameters

We have three hyper-parameters in our method:(1) δ: the threshold for l_2 distance between two words' embeddings. (2) β: the threshold for selecting suspects in the corrector. (3) α: the threshold in the frequency-based detector.

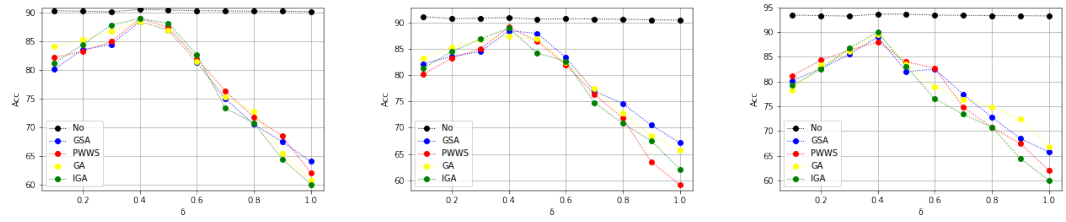
As shown in Figure 2, the accuracy on benign sentences (generalization) remains nearly unchanged, and we only need to select the hyper-parameters based on accuracy on adversarial sentences (robustness). Then, we investigate how the distance δ influences the performance of TextShield. When $\delta = 0$ or $\delta = 1$, the performance is not satisfactory because a small δ results in insufficient word synonyms, and a large δ may include words that are not semantically close. When $\delta < 0.5$, the performance increases continuously as δ increases. When δ exceeds 0.5, the accuracy starts to decrease rapidly. Therefore, we choose $\delta = 0.5$ to preserve the best robustness against text attacks.

Similarly, we study the impact of β in the corrector. As shown in Figure 3, the performance achieves the best when β is 0.4, and either a small or large β will cause a significant performance drop, especially when β is large. This is because β serves to control the ratio of suspects in a sentence: when β is 0, then only the most important word is considered as a suspect, which may be inadequate; on the contrary, when β is 1.0, then every word will be viewed as a suspect, which is inappropriate.



(a) TextCNN (b) LSTM (c) Bert

Figure 2: Accuracy (%) of TextShield as δ varies from 0.1 to 1.0 for three victim models on AG's News, δ is set as 0.5.



(a) TextCNN (b) LSTM (c) Bert

Figure 3: Accuracy (%) of TextShield as β varies from 0.1 to 1.0 for three victim models on AG's News, β is set as 0.4.

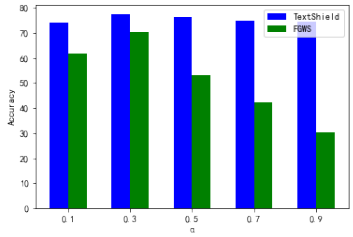


Figure 4: Accuracy (%) of TextShield and FGWS on adversarial sentences as α changes.

Finally, the previous detection-based method (i.e., FGWS) is sensitive to the frequency difference. FGWS remains sensitive to the threshold α. Therefore, we compare the performance of TextShield and FGWS as α changes. As Figure 4 shows, TextShield achieves better and more stable performance than FGWS as α changes, especially when α is getting larger. Such results indicate that TextShield has less dependence on word frequency due to the support of gradient-based detectors.

5.2 Analysis of ensembling

Since a crucial design of TextShield is the combination of various detectors to judge whether the input sentence is an adversarial one, we investigate the impact of such an ensembling strategy. Therefore, we verify the performance after removing some detectors, and the evaluation is launched on TextCNN on AGNews.

Table 5: Accuracy (%) drop after removing some detectors. '-VG' means removing the VG detector in TextShield.

Removed detector	GSA	PWWS	GA	IGA
-VG detector	-8.9	-7.9	-6.4	-7.0
-IG detector	-3.8	-4.5	-2.0	-3.5
-LRP detector	-4.6	-4.0	-1.8	-4.0
-GBP detector	-3.4	-2.5	-1.1	-2.7
-Freq detector	-5.0	-4.1	-3.3	-4.8
-All detectors	-34.8	-29.6	-40.3	-36.9

As shown in Table 5, removing any detector will cause a performance drop to TextShield, demonstrating the effectiveness of combining various detectors with an ensemble mechanism. Specifically, based on the results, the importance of detectors can be ranked as follows: VG>Freq>LRP>IG>GBP. The results confirm our arguments that such different detectors can complement well with each other. Furthermore, removing all the detectors shows a catastrophic performance drop, which strongly shows the importance of the integrity of the detector-corrector pipeline.

6 CONCLUSION

In this paper, we propose TextShield, a detection-correction pipelined ensemble defense against word-level text attacks. We firstly design a model-specific, task-specific, and sentence-specific word importance through gradient information of DNN-based models. TextShield leverages the word importance to build detectors and uses the random forest to combine various detectors, which further improves the

robustness of detectors. Finally, the corrector rectifies the adversarial sentences into benign ones and feeds them to the classifier. Comprehensive experiments demonstrate that TextShield is superior to SoTA baselines in defending against text attacks and their transferability. Since such a pipeline is not specific to text classification, we will explore the capacity of TextShield on other tasks (e.g., QA) in the future.

REFERENCES

[1] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating Natural Language Adversarial Examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2890–2896.

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10, 7 (2015), e0130140.

[3] Samuel Barham and Soheil Feizi. 2019. Interpretable adversarial training for text. *arXiv preprint arXiv:1905.12864* (2019).

[4] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. 2016. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*. Springer, 63–71.

[5] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[7] Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2020. Towards robustness against natural language word substitutions. In *International Conference on Learning Representations*.

[8] Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. Towards robustness against natural language word substitutions. *arXiv preprint arXiv:2107.13541* (2021).

[9] Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based Adversarial Examples for Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6174–6181.

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

[11] Chuan Guo, Jacob Gardner, Yurong You, Andrew Wilson, and Kilian Weinberger. 2019. Simple Black-box Adversarial Attacks. *Proceedings of Machine Learning Research* (2019).

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[13] 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. 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)*. 4083–4093.

[14] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified Robustness to Adversarial Word Substitutions. 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)*. 4129–4142.

[15] Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust Encodings: A Framework for Combating Adversarial Typos. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2752–2765.

[16] Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* (1972).

[17] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1746–1751.

[18] Ching-Yun Ko, Zhaoyang Lyu, Lily Weng, Luca Daniel, Ngai Wong, and Dahua Lin. 2019. POPQORN: Quantifying robustness of recurrent neural networks. In *International Conference on Machine Learning*. PMLR, 3468–3477.

[19] Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems. (2018).

[20] Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and William B Dolan. 2021. Contextualized Perturbation for Textual Adversarial Attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 5053–5069.

[21] Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development* 2, 2 (1958), 159–165.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

[23] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. 2019. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning* (2019), 193–209.

[24] Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-Guided Word Substitutions for Detecting Textual Adversarial Examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 171–186.

[25] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 372–387.

[26] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[27] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[28] Christopher Potts. 2010. On the negativity of negation. In *Semantics and Linguistic Theory*, Vol. 20. 636–659.

[29] Xiaojun Quan, Liu Wenyin, and Bite Qiu. 2010. Term weighting schemes for question categorization. *IEEE transactions on pattern analysis and machine intelligence* 33, 5 (2010), 1009–1021.

[30] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified Defenses against Adversarial Examples. In *International Conference on Learning Representations*.

[31] 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*. 1085–1097.

[32] Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812* (2017).

[33] Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 4323–4330.

[34] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge.

[35] Imran Sheikh, Irina Illina, Dominique Fohr, and Georges Linares. 2016. Learning word importance with the neural bag-of-words model. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. 222–229.

[36] Zhouxing Shi, Huan Zhang, Kai-Wei Chang, Minlie Huang, and Cho-Jui Hsieh. 2019. Robustness Verification for Transformers. In *International Conference on Learning Representations*.

[37] J Springenberg, Alexey Dosovitskiy, Thomas Brox, and M Riedmiller. 2015. Striving for Simplicity: The All Convolutional Net. In *ICLR (workshop track)*.

[38] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*. PMLR, 3319–3328.

[39] Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2020. InfoBERT: Improving Robustness of Language Models from An Information Theoretic Perspective. In *International Conference on Learning Representations*.

[40] Wenqi Wang, Run Wang, Lina Wang, Zhibo Wang, and Aoshuang Ye. 2019. Towards a robust deep neural network in texts: A survey. *arXiv preprint arXiv:1902.07285* (2019).

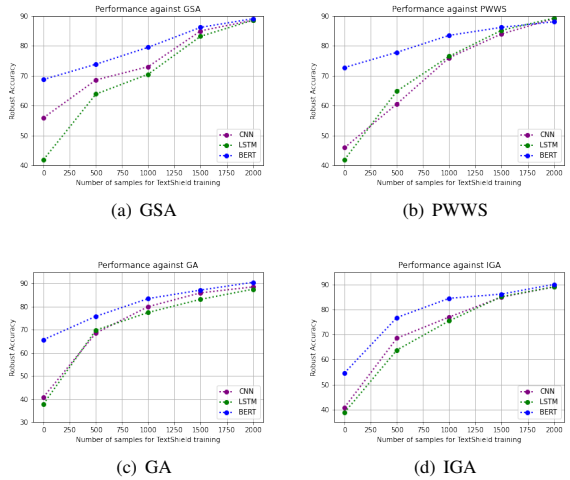
[41] Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural Language Adversarial Attack and Defense in Word Level. (2019).

[42] Xiaosen Wang, Hao Jin, Yichen Yang, and Kun He. 2021. Natural Language Adversarial Defense through Synonym Encoding. In *Conference on Uncertainty in Artificial Intelligence*.

[43] Eric Wong and Zico Kolter. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*. PMLR, 5286–5295.

[44] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. 2020. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems* 33 (2020).

[45] Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level Textual Adversarial Attacking as Combinatorial Optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6066–6080.



(a) GSA (b) PWWS

(c) GA (d) IGA

Figure 5: Accuracy (%) of TextShield on adversarial sentences generated by four attacks as K changes.

[46] Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. Openattack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. 363–371. https://doi.org/10.18653/v1/2021.acl-demo.43

[47] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*. PMLR, 7472–7482.

[48] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems* 28 (2015), 649–657.

A SETTINGS OF TOY EXPERIMENTS

A.1 Setting of toy experiment 1

The dataset is AGNews [48], and we apply three popular text attacks (GSA, PWWS, and GA) to generate adversarial sentences. Specifically, we choose 300 adversarial sentences that successfully mislead the classifier; thus, the original attack success rate is 100%. Then, for each of these 300 sentences, we calculate each word's importance by the vanilla gradient. Then, we select words with top 10% highest word importance, replace them with [MASK], and feed the partially masked sentence to the classifier for evaluation.

A.2 Setting of toy experiment 2

Similar to toy experiment 1, the dataset is AGNews [48], and we apply three popular text attacks (GSA, PWWS, and GA) to generate adversarial sentences. Still, we choose 300 adversarial sentences that successfully mislead the classifier. Then, for each sentence from these 300 sentences, we calculate each word's frequency among the dataset. Then, we select words with top 10% highest word importance and replace them with the most frequent synonym. Specifically, we regard two words as synonyms if their word embedding $\mathbf{w_i}$, $\mathbf{w_j}$ satisfy that $||\mathbf{w_i} - \mathbf{w_j}|| \leq 0.5$. Finally, we feed replaced sentences to the classifier for evaluation.

B ANALYSIS OF THE NUMBER OF ADVERSARIAL SENTENCES USED IN TEXTSHIELD TRAINING

During the training process, we use the same number K of adversarial sentences for TextShield. The results in the main paper are under the condition where $K = 2000$. This section illustrates the relation between TextShield and K.

Specifically, we keep the same training settings mentioned in the main paper except for K. The performance of TextShield under different K is shown in Figure 5.

As Figure 5 shows, the defense capacities of TextShield are enhanced as K continuously increases since more samples in the training help the detectors recognize adversarial sentences better. Specifically, when K is set as 500, TextShield still significantly outperforms the performance of AT when using 2000 adversarial sentences for training, further validating the effectiveness of TextShield.