WIDDAS: A Word-Importance-Distribution-based Detection method against Word-Level Adversarial Samples

Xiangge Li, Hong Luo* and Yan Sun

Beijing University of Posts and Telecommunications, Beijing 100876, China
luoh@bupt.edu.cn

Abstract. Deep neural networks are facing security threats from adversarial samples, and even the most advanced large-scale language models are still vulnerable to adversarial attacks. Moreover, existing defense methods against adversarial attacks suffer from issues such as low accuracy in detection, too much false detection of clean data, and high defense costs. Therefore, in this paper, we propose WIDDAS: a Word-Importance-Distribution-based Detection method against Word-Level Adversarial Samples . It comprises a detection module and an evaluation module. The detection module swiftly identifies potential adversarial samples based on the word importance distribution of the input text. Then the evaluation module attempts to restore those samples and evaluates whether they are adversarial, thereby filtering out clean data which is non-adversarial. Experimental results demonstrate that WIDDAS outperforms the baselines in terms of both detection accuracy for adversarial samples and clean da-ta. Particularly in scenario of Chinese data, the detection accuracy is at least 4.5% higher than the best baseline.

Keywords: Nature Language Processing, Adversarial Samples, Textual Defense, Adversarial Detection, Model Robustness.

1 Introduction

In recent years, Deep Neural Networks (DNNs) have achieved state-of-the-art performance in numerous Natural Language Processing (NLP) tasks [1]. However, several studies highlighted the vulnerability of DNNs to adversarial samples [2], which pose a significant threat to their robustness. Attackers make the model output wrong result through modifying only few keywords in the input text, while human can still understand its original semantic. This method is commonly referred to as word-level adversarial attack, and the modified text is known as adversarial sample.

Currently, even the state-of-the-art Large-scale Language Models (LLMs) remain vulnerable to adversarial attacks [3][4]. Consequently, the issue of adversarial robustness has draw widely attention. The defense method against adversarial samples become a prominent topic in NLP field. It can be categorized into two types: detection

defense and complete defense. 1) **Detection defense** aims to identify whether the input text is an adversarial example. A notable method is additional networks [5]. 2) **Complete defense** aims at making the model correctly classify the input text, without detecting whether it is an adversarial sample. The representative work is adversarial training [6] and redesigning networks [7]. Most previous work enhances the adversarial robustness for small-scale NLP models through complete defense.

LLMs are increasingly being integrated into a variety of NLP tasks. Nevertheless, it is hard to implement complete defense for LLMs due to these factors: 1) **High cost of adversarial training**. LLMs have a large number of parameters, and hence rendering the cost higher. 2) **Limited accessibility for modification of network**. Many LLM providers only offer API service, thereby users who buy LLM service for deployment cannot improve the adversarial robustness through redesigning the network. 3) **Wastage of computational resources**. LLMs have a higher computational cost, so the adversarial attacks result in a wastage of computational resources. Since complete defense cannot detect and block attacks, we believe that detection defense is more suitable to counter the adversarial threat faced by LLMs.

Current detection methods against word-level adversarial attacks can be divided into three types: textual feature-based detection, context-based detection, and victim model-based detection. ***Textual feature-based detection*** primarily extracts features such as part of speech, perplexity, and text length. Then a supervised classifier is trained based on them to identify adversarial inputs. ***Context-based detection*** leverages contextual information within the text to locate perturbations and subsequently restores the text. ***Victim model-based detection*** generates multiple samples with word replacement. It determines the adversary based on the consistency of output labels from the victim model. The drawbacks of there methods are as follows: 1) Textual features-based and context-based methods only consider the abnormal features in inputs, without considering the adversarial impact on the victim model based on its output labels. Consequently, it may lead to false detection of clean data or texts with non-adversarial spelling errors. 2) Victim model-based method relies heavily on output labels from the victim model. So, it is impractical in defensive scenarios with limited computation resources, especially when the victim model is a LLM. Therefore, the challenge lies in accurately detecting adversarial samples based on textual features, while minimizing dependence on the output labels of LLMs and reducing false detection rates.

To address above issues, we propose a detection method against word-level adversarial samples based on the distribution of word importance (WIDDAS). Through considering both textual features and the adversary of inputs to victim model, WIDDAS enhances detection accuracy on both clean data and adversarial samples. Rather than relying on output labels from the victim model, we introduce a textual entailment model to evaluate the adversary. Our main contributions are as follows:

- We propose a general defense method against word-level adversarial samples. It does not depend on outputs result from the victim model, therefore it is easily adaptable to various models.
- We incorporate the distribution of word importance as detection features to quickly evaluate potential adversarial samples based on the language model. Consequently,

our method reduces the false detection rate both on clean data and text with non-adversarial spelling errors. It holds significant potential in the field of adversarial tracing and mitigation.

- Our method demonstrates broadly applicability across multiple languages, especially on Chinese data with various and complex attack methods.

2 Related Work

2.1 Textual Attack

Most of the current research focuses on textual attack method in black-box scenarios, where attackers can only obtain the output of victim model. According to the different strategies to generate adversarial examples, we could divide it into three categories [8]: 1) Character-level attacks. 2) Word-level attacks. 3) Sentence-level attacks. Among them, word-level attack is the predominant method, employing keyword replacement to achieve perturbation, such as [9][10][11]. In addition, some studies concentrate on low-resource language attack methods, such as Chinese [12][13][14]. Due to its rich linguistic features, these methods employ diverse strategies for generating adversarial sample in Chinese, such as homophones, visually similar characters, splitting characters. Therefore, detecting adversarial samples in Chinese becomes considerably more challenging.

2.2 Textual Complete Defense

In order to defend against textual attacks, most early works focus on complete defense. One of the common techniques is adversarial training, such as [15][16][17]. It solves the sparsity problem of samples by adding adversarial samples to the training data, and then improves the adversarial robustness of the model. However, Du et.al [17] pointed out that, adversarial training cannot fully simulate the real input space and is only robust to the constructed adversarial attacks. YOO et.al found that [18], adversarial training may lead to a decline in the accuracy on the models' classification of clean data.

Another method of complete defense is to resign the DNN and optimize its structure. Jones et.al proposed the RobEn model [19], which mapping the Embedding of the input sentences to a smaller discrete coding space and then retraining the whole model for prediction. However, the prediction accuracy of the modified model decreases. Sun et.al [20] proposed a robust ChineseBERT model for Chinese data, which extract four multimodal information including semantic, glyphs and phonemes. It improves the model robustness based on multimodal fusion vectors. Similarly, Su et al. presented RoCBert [7], which is a pretrained Chinese BERT with Multimodal Contrastive Pretraining. RoCBert is robust to various forms of adversarial attacks like word perturbation, synonyms, typos, etc. Nevertheless, these defense methods require retraining the DNN model, rendering them unsuitable for models with high training costs like LLMs.

2.3 Textual Detection Defense

Detection defense commonly trains an additional network to identify adversarial samples. One prevalent approach is textual feature-based detection. For example, Alon et.al trained a detector based on LightGBM [21], which extracting features including text length and perplexity. Furthermore, Zhu et.al [4] pointed out that adversarial attacks may lead to attention divergence for input text, which serves as a potential basis for detection. However, it is a challenge to obtain the word importance score in black-box defense scenarios.

Another method is context-based detection. For instance, Zhou et.al proposed DISP [22], which trained a perturbation discriminator to identify adversarial attacks. DISP can validates how likely a token in text is perturbed and provides a set of potential perturbations based on contextual Embedding.

Moreover, there is a victim model-based defense method. For instance, Wang et.al believe that adversarial samples mislead the classifier through changing the interaction between words, and accordingly proposed RS&V [23]. It generates a new set of samples with word replacement based on synonyms and obtain the victim model's output labels. When those labels are inconsistent with the original text, it identifies the input text as a adversarial sample. However, due to its reliance on the output labels of victim model, this approach may not be suitable with limited computational resources, especially when the victim model is a LLM.

3 Methodology

3.1 Framework

As shown in Figure 1, the framework for WIDDAS consists of a detection module and an evaluation module. Initially, the input text is processed by the detection module to swiftly identify the potential adversarial samples based on the distribution of Word Importance Score (WIS). Those samples might include both real adversarial samples and a few clean data. Subsequently, the evaluation module filters those samples and retains only adversarial samples.

3.2 Detection Module

The clean texts typically exhibit efficient attention allocation, where a limited number of keywords have high word importance score. In contrast, the adversarial attack leads to attention divergence [4]. Therefore, we design a detection module to leverage this observation. It comprises a Word Importance Ranking Model and a detection model based on the distribution of word importance. The two models require pretraining process before detection.

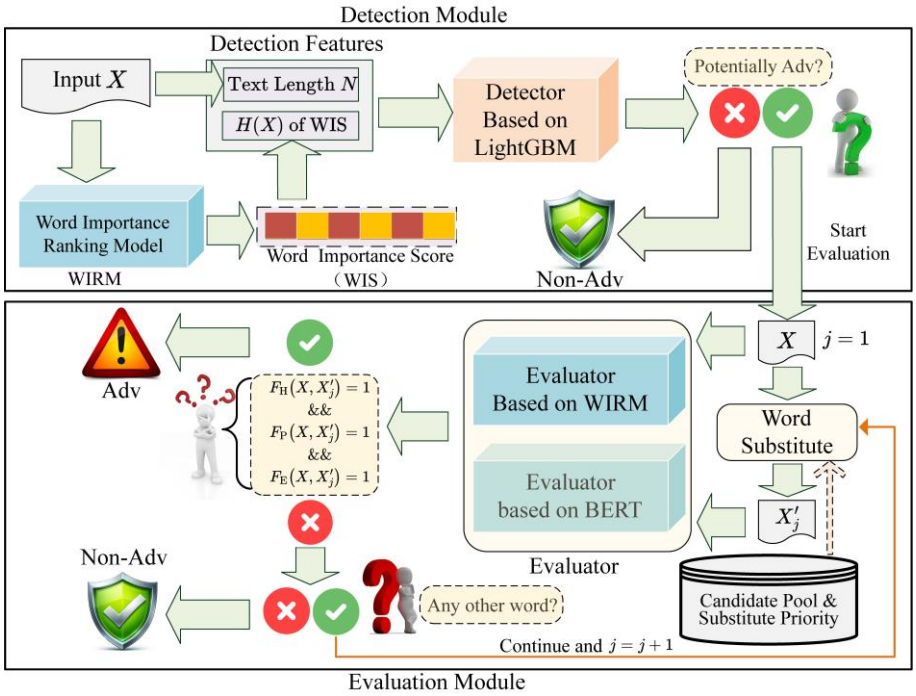

Fig. 1. The Framework of WIDDAS

Word Importance Ranking Model (WIRM). The adversarial transferability principle suggests that adversarial samples generated for a substitute model f' may also be adversarial against the victim model f. This is due to different victim models heavily concern on similar keywords when extract the semantic. Therefore, in order to detect the adversarial samples in black-box scenarios, we can train a Word Important Ranking Model (WIRM), which aims to obtain the distribution of WIS for input text, based on the adversarial transferability. We adopt a BiGRU network combined with attention mechanism as the backbone of WIRM. In this setup, the attention weight serves as the WIS. The detailed structure of WIRM is shown in Figure 2. Please note that the first paragraph of a section or subsection is not indented.

Firstly, the input text $X = (x_1, \ldots x_i, \ldots, x_N)$ is encoded by BiGRU to obtain the hidden state $H = (h_1, \ldots h_i, \ldots, h_N)$. The i-th hidden state h_i is computed as:

$$h_i = \mathrm{BiGRU}(x_i, h_{i-1}) \tag{1}$$

Then the attention matrix $\alpha = (\alpha_1, \ldots, \alpha_i, \ldots, \alpha_N)$ is calculated with H_s. The α_i can be formulated as:

$$\alpha_i = \frac{\exp(u_i^{\mathrm{T}} u_i)}{\sum_{j=1}^{N} \exp(u_j^{\mathrm{T}} u_j)} \tag{2}$$

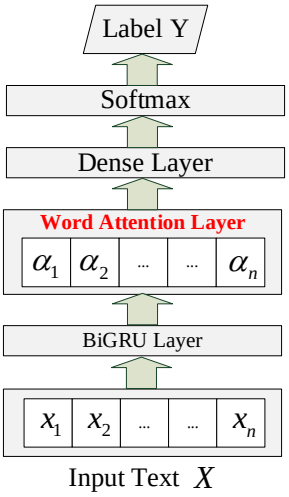

Fig. 2. The structure of WIRM

$$u_i = \tanh(W_{\text{word}}\, h_i + b_{\text{word}}) \tag{3}$$

where W_{word} is the weight of a one-layer MLP, b_{word} represents the bias, and u_i is a hidden state of h_i. The final vector s of the input text X is denoted as follows:

$$s = \sum_{i=1}^{N} \alpha_i u_i \tag{4}$$

After a fully connected layer and a softmax layer, the probability distribution p for classification is obtained. Then we use the negative log likelihood of the correct labels as training loss L. They can be formulated as:

$$p = \text{softmax}(W_{\text{fc}} \cdot s + b_{\text{fc}}) \tag{5}$$

$$L = \sum_{k} \log p_{kj} \tag{6}$$

where W_{fc}represents the weight of fully connected layer, b_{fc} is the bias, k is the number of input text, j is the label of X. Before detection, WIRM is initially trained with task-specific supervised data. During the detection process, we employ the pretrained WIRM to obtain the WIS, donated as α. Finally, we calculate the entropy of WIS, denoted as $H(X)$ to represent its distribution. $H(X)$ is formulated as:

$$H(X) = \text{Entropy}(\alpha) \tag{7}$$

$$\alpha = \text{WIRM}(X) \tag{8}$$

Detection Model based on the distribution of word importance. The target of the detection model is to swiftly identify potential adversarial samples. We adopt a LightGBM model as detector, which is an ensemble learning algorithm based on decision trees. It has high efficiency and low space cost. The input features consist of

$H(X)$ from WIRM and the text length. The output result, denoted by Y, can be formulated as:

$$Y = \text{LightGBM}(H(X), N) \tag{9}$$

where $Y \in \{0, 1\}$. and $Y = 1$ when X is adversarial.

The detector requires pretrain. We employ Textfooler [6], HLBB [11] and PSO [24] to generate English adversarial samples, while CWordCheater [12], CWordAttacker [14], WordHanding [13] to generate Chinese adversarial samples. Then the clean data and adversarial samples are mixed to conducted the training dataset.

3.3 Evaluation Module

Since the potential adversarial samples output from the detection module might residue a few amount of non-adversarial samples, including 1) Clean data. 2) Texts with non-adversarial spelling errors, such as typos resulting from OCR or speech recognition. To minimize the false detection rate on these samples, we design an evaluation module to further filter out and so retain only the adversarial samples.

The evaluation module operates through a three-step process: constructing candidate pool for word substitution, calculating the priority of replacement, and constructing word substitution followed by an evaluation. Firstly, we select keywords based on the WIS and replace them with candidate pool, aiming to restore the original text. Then the replaced text is evaluated with BERT and WIRM models to identify its adversary.

Candidate Pool. To construct candidate pool for different languages, we can employ commonly utilized adversarial attacks. For instance, in the case of English, a synonym dictionary can be used for candidate pool. In the case of Chinese, all homophones, visually similar characters and splitting characters dictionaries can be included in the candidate pool.

Priority of Replacement. In the case of English text, each token within a sentence is set to '[Mask]' in the order of WIS. Then BERT determine their scores with Masked Language Model tasks. We posit that tokens with lower scores are more likely being perturbed and thus possess a higher priority for replacement. To expedite this process, only the tokens within the lowest 30% percentile will be substituted.

Comparing with English, Chinese attack usually generates adversarial samples with an extra method, which splitting a single character into two independent characters based on left-right structure. Therefore, for Chinese text, we firstly refer to the splitting dictionary to identify whether or not the adjacent characters can be merged into a single character, and give them the highest priority for replacement. Then we calculate priorities for remaining text, similar to English.

Substitution and Evaluation. We use three indicators, including the distribution of WIS, perplexity, and textual entailment result to verify the adversary of the replaced text X_j'.

- Indicator based on distribution of WIS, $F_{\mathrm{H}}(X, X_j')$. Considering that adversarial samples exhibit attention divergence, we calculate the distribution of WIS, including $H(X)$ and $H(X_j')$. If X is adversarial, a correctly replaced X_j' should satisfy $H(X) > H(X_j')$. Therefore, $F_{\mathrm{H}}(X, X_j')$ can be denoted as:

$$F_{\mathrm{H}}(X, X_j') = \begin{cases} 1 & H(X) > H(X_j') \\ 0 & \text{else} \end{cases} \tag{10}$$

- Indicator based on perplexity, $F_{\mathrm{P}}(X, X_j')$. Perplexity can evaluate the fluency and quality of the sentence in adversarial detection. An adversarial sample with modified perturbation may result in high perplexity values compared to its original form. The perplexity of input text X is calculated as follows:

$$PPL(X) = P(x_1, x_2, \dots, x_N)^{\frac{1}{N}} \tag{11}$$

  where $P(x_1, x_2, \dots, x_N)$ represents the probability assigned by the language model for predicting the occurrence of X. Thus, we calculate the perplexity of X_j' and X. If X is adversarial, the correct replaced X_j' should satisfy $PPL(X) > PPL(X_j')$. $F_{\mathrm{P}}(X, X_j')$ can be denoted as:

$$F_{\mathrm{P}}(X, X_j') = \begin{cases} 1 & PPL(X) > PPL(X_j') \\ 0 & \text{else} \end{cases} \tag{12}$$

- Indicator based on Textual Entailment, $F_{\mathrm{E}}(X, X_j')$. Recognizing Textual Entailment (RTE) is an important task in the field of NLP. It aims to determine the relationship between two pieces of text, including entailment, contradiction and neutral. We introduce an RTE model to verify the adversary of X. When X and X_j' exhibit contradiction or neutral (i.e., $RTE(X, X_j') = 0$), it indicates the semantic may be changed, X is highly possible to be adversarial. $F_{\mathrm{E}}(X, X_j')$ is calculated as follows:

$$F_{\mathrm{E}}(X, X_j') = \begin{cases} 1 & RTE(X, X_j') = 0 \\ 0 & \text{else} \end{cases} \tag{13}$$

Before evaluation stage, it is necessary to generate X_j' through word substitution. Firstly, the set of substituted words W is conducted based on WIS. Then we replace each w_k in W in order by priority. It is done iteratively and returns an optimal X_j'. In each round of substitution, the algorithm of $\mathrm{WordSubstitute}(X, X_{j-1}', w_k, C_k)$ is as follows:

Algorithm 1 WordSubstitue

Require: Input X, the j-th input X_{j-1}', the replaced word w_k, candidate pool C_k

Ensure： The optima result X_j'

1:$\mathrm{H}_{\mathrm{best}} = +\infty$; $\mathrm{RTE}_{\mathrm{best}} = 1$;

2:for c_k^i in C_k **do**

3: $X_t' = \mathrm{Replace}(X_{j-1}', w_k, c_k^i)$;

4: if $RTE(X, X_t') = 0$ **and** $\mathrm{RTE}_{\mathrm{best}} = 1$ **then**

5: $X_{\mathrm{best}} = X_t'$; ▷Firstly consider the text entailment relationship

6: $\mathrm{H}_{\mathrm{best}} = H(X_t')$;

7: $\mathrm{RTE}_{\mathrm{best}} = RTE(X, X_t')$;

8: else if $RTE(X, X_t') = \mathrm{RTE}_{\mathrm{best}}$ **and** $H(X_t') < \mathrm{H}_{\mathrm{best}}$ **then**

9: $X_{\mathrm{best}} = X_t'$;

10: $\mathrm{H}_{\mathrm{best}} = H(X_t')$; ▷Select a sample with lower distribution of WIS

11: **end if**

12:$X_j' = X_{\mathrm{best}}$

13:**return** X_j'

3.4 The Whole Algorithm

We believe that, for an adversarial sample X, its correct replacement X_j' should satisfy the above three evaluations, which means the distribution of WIS decreases, the perplexity value decreases, and the relationship is contradiction or neutral. The whole detection algorithm is shown as Algorithm 2.

Algorithm 2 The Whole Detection Process

Require： Input X, Length N, Candidate Pool C, Word Importance Ranking Model WIRM.

Ensure： The adversary of X

1:$\alpha = \mathrm{WIRM}(X)$; ▷Obtain the WIS of X

2:$H(X) = \mathrm{Entropy}(\alpha)$; ▷Obtain the Entropy of WIS

3:$Y = \mathrm{LightGBM}(H(X), N)$ ▷Detect based on LightGBM

4:**if** $Y = 0$ **then**

5: **return** non-adversarial ▷X is non-adversarial, end

6:**else**

7: Construct W based on α; ▷The set of substitute word in X

8: $j = 1, X'_0 = X$;

9: **for** w_k in W **do**

10: Construct C_k for w_k from C; ▷The candidate pool for w_k

11: $X'_j = \text{WordSubstitute}(X, X'_{j-1}, w_k, C_k)$; ▷The global optimal X'_j

12: $a'_j = \text{WIRM}(X'_j)$;

13: $H(X'_j) =$ Entropy (a'_j);

14: Calculate $F_{\text{H}}(X, X'_j)$ with $H(X)$ and $H(X'_j)$;

15: Calculate $F_{\text{P}}(X, X'_j)$ with $PPL(X)$ and $PPL(X'_j)$;

16: Calculate $F_{\text{E}}(X, X'_j)$ with $RTE(X, X'_j)$;

17: **if** $F_{\text{H}}(X, X'_j) = 1$ **and** $F_{\text{P}}(X, X'_j) = 1$ **and** $F_{\text{E}}(X, X'_j) = 0$ **then**

18: **return** adversarial ▷X is adversarial

19: **end** if

20: **end for**

21:**end if**

22:**return** non-adversarial

4 Experiments

Take the text classification task as example, we conduct extensive experiments on three English datasets validate the effectiveness of our method. Furthermore, in order to validate the applicability on different languages, we also evaluate our method on three Chinese datasets.

4.1 Experiments Setup

Datasets. We adopt three English datasets including IMDB [25], Yelp [26], MR [27] and three Chinese datasets including Waimai, OnlineShopping and Hotel from ChnSentiCorp for text classification. Then we generate and mix adversarial samples with them to conduct evaluation datasets

Adversarial Samples. Firstly, we adopt BERT and ERNIE as the victim model for English and Chinese data, respectively. Then we attack the BERT with Textfooler [6],

HLBB [11] and PSO [24], and the ERNIE model was attacked with CWordAttacker (CWA) [14], CWordCheater (CWC) [12] and WordHanding [13]. When the generated sample attacks the victim model successfully, we save it as adversarial sample.

Baselines. We take three detection defense methods DPPL [21], RS&V [23] and DISP [22] as our baselines. DPPL detects adversarial samples based on perplexity and text length. RS&V identifies adversarial samples through random synonym substitution and logit-based voting. DISP trains a detection network to locate the wrong words based on context.

4.2 The Evaluation of Defense Effectiveness on Classical Language Models

We first conduct evaluations on classical Language Models, using the above datasets. The evaluation metrics include the accuracy of clean data (Clean Acc, %) and the accuracy of adversarial samples (Adv Acc, %). We adopt BERT and ERNIE as the victim model. Table 1 and Table 2 show the results on English datasets and Chinese datasets, respectively. We can see that:

Table 1. The detection effectiveness on BERT for English datasets

Dataset	Method	Clean Acc	Adv Acc		
			Textfooler	HLBB	PSO
IMDB	DISP	96.7	63.4	69.2	66.1
	DPPL	95.4	71.3	74.1	70.5
	RS&V	98	**88.7**	87.9	82.7
	WIDDAS	**98.7**	88.5	**89.6**	**85.4**
Yelp	DISP	97.4	62.6	67.9	63
	DPPL	94.8	69.2	72.4	69.8
	RS&V	98.6	85.2	85.3	84.7
	WIDDAS	**98.9**	**86.1**	**87.4**	**87**
MR	DISP	97.5	63.8	66.5	63.2
	DPPL	95.2	72.9	75	71.3
	RS&V	98.3	86	**87.3**	86.4
	WIDDAS	**98.4**	**86.3**	87.2	**86.8**

- The detection accuracy of DPPL and DISP on English clean data is poor. This is because they only consider the context and textual features without the adversary to victim model. RS&V partially decreases false detections through refer to output labels from the victim model. WIDDAS not only considers textual features but also introduces a RTE model to verify whether the text semantic changed. Thus, WIDDAS achieves superior detection performance on clean data with an accuracy exceeding 98.4%. Moreover, compared to RS&V, WIDDAS does not rely on the output labels from victim model, resulting in lower computational costs.

- The four methods show significant disparities in accuracy when detecting adversarial samples in English. DISP performance the worst as it heavily relies on contextual representation for detection. And attackers often add constraints of contextual Embedding against detection algorithm, which leads to poor detection accuracy of DISP. Similarly, DPPL, which detects based on perplexity, also shows lower accuracy. This is because that attackers ensure the quality of generated text with semantical and grammatical constraints, thereby misleading DPPL. RS&V identifies a large amount of adversarial samples through synonym replacement and voting. WIDDAS accurately capturing attention divergence caused by adversarial attacks. Therefore, it performances best in most scenarios with an accuracy rate exceeding 85.4%.

Table 2. The detection effectiveness on ERNIE for Chinese datasets

Dataset	Method	Clean Acc	Adv Acc		
			CWC	CWA	WordHanding
Hotel	DISP	96.8	64.3	68.1	70.1
	DPPL	94.6	72.8	75.3	73.7
	RS&V	97.1	80.6	81.2	84.5
	WIDDAS	**97.9**	**86.4**	**87.1**	**89.2**
shopping	DISP	95.3	65.9	67	68.4
	DPPL	94.2	73.1	76	75.3
	RS&V	96.7	81.5	81.9	83.9
	WIDDAS	**97.6**	**85**	**86.7**	**88.6**
Waimai	DISP	95.9	65.2	67.4	68
	DPPL	95	73.9	76.5	73.8
	RS&V	96.4	81.7	82.3	84.1
	WIDDAS	**97.2**	**86.7**	**87.2**	**89.5**

- In Chinese detection scenario, both DISP and DPPL shows some improvement in accuracy. But they still remain significantly lower compared to RS&V and WIDDAS. The gap between RS&V and WIDDAS has widened even further, with a minimum difference of 0.8% in detection accuracy on clean data and at least 4.5% in adversarial detection accuracy. It claims that detection based on distribution of word importance is more effective against various types of attacks in Chinese scenarios. Additionally, WIDDAS constructs a replacement candidate pool based on Chinese language features, making it easier to restore the correct text. Thus, WIDDAS filters out more non-adversarial samples.

4.3 The Adversarial Detection Effectiveness on the LLM

Then, we evaluate adversarial detection effectiveness on the LLM (ChatGLM3- 6B [28]). Table 3 show the results. Compared to classical language models, the adversarial

detection accuracy for all four methods decrease when targeting LLM. We believe this is because that the better robustness of the LLM, which means that adversarial samples against the LLM are well camouflaged and harder to detect. However, WIDDAS still maintains at least 76.1% accuracy rate, the best among the four methods. It suggests that it is equally effective for adversarial detection on LLMs.

Table 3. The adversarial detection effectiveness on ChatGLM3-6B

Language	Dataset	Attack Method	Adv Acc			
			DISP	DPPL	RS&V	WIDDAS
English	IMDB	Textfooler	60.2	67.1	80.3	**82.7**
		HLBB	64.8	71.4	83	**84.9**
		PSO	58.5	65.3	77.6	**81.4**
	Yelp	Textfooler	55.4	62.7	76.1	**79.3**
		HLBB	59.6	63.8	81.2	**83.5**
		PSO	61	64.5	80.9	**82.5**
	MR	Textfooler	57.9	59.4	73.6	**76.1**
		HLBB	58.1	60.9	81.4	**83.2**
		PSO	56.7	62.5	82.8	**84**
Chinese	Hotel	CWC	59.7	65.3	73.4	**78.2**
		CWA	62.8	68.5	76.1	**80.6**
		WordHanding	67.4	69.2	79	**83**
	Shopping	CWC	61.5	66	78.2	**81.4**
		CWA	63.3	70.2	79.5	**82.8**
		WordHanding	64.1	72.8	82.7	**83.9**
	Waimai	CWC	62	70.6	78.3	**83.1**
		CWA	64.2	73	80.1	**81.9**
		WordHanding	66.9	73.5	81.6	**84.8**

4.4 Evaluation on Texts with Non-adversarial Misspellings

Users may occasionally input text with non-adversarial spelling errors. These include characters spelling errors in English, and phonetic or visually spelling errors in Chinese caused by speech recognition input methods and OCR. An effective detection method is supposed to distinguish these as non-adversarial samples. To evaluate the false detection rates of above methods, we introduce non-adversarial samples into above six datasets. For English datasets, we randomly select few words and adds character-level perturbations like insertions, deletions or exchanges. For Chinese datasets, we randomly select few characters and replace them with homophones or visually similar characters from dictionary. These modified samples are sent to victim model. If the output label remain unchanged, we save it as text with non-adversarial spelling errors. Table 4 shows the detection accuracy on these English and Chinese data.

It is obviously that DISP and DPPL exhibit lower accuracy, indicating a higher false detection rate. Particularly, the perplexity-based DPPL shows the lowest accuracy as the spelling errors tend to increase the perplexity value. In contrast, WIDDAS outperformances the other methods, especially in Chinese scenario. It suggests that the detection method with multiple evaluation strategies effectively filters out texts with non-adversarial spelling errors and reduce false detection.

Table 4. The defense effectiveness of texts with non-adversarial misspellings

Detection Method	English Acc (%)			Chinese Acc (%)		
	IMDB	Yelp	MR	Hotel	Waimai	Shopping
DISP	85.1	86.9	85.4	83.2	83.6	84.1
DPPL	84.3	85.7	83	80.4	81.3	79.6
RS&V	94.6	94.2	95.8	89.7	88.5	88.3
WIDDAS	**97.8**	**97.5**	**96.7**	**96.5**	**95.8**	**97**

4.5 Ablation Experiment

We conduct the ablation experiments to verify the effects of Evaluate Module. The evaluation metrics include the accuracy of clean data (Clean Acc, %) , the accuracy of text with non-adversarial spelling errors (SpeErr Acc, %), and the accuracy of adversarial samples (Adv Acc, %). We first use the detection module alone (Dec), then combined it with evaluation module (Dec+Eva). Table 5 shows the result of attacking BERT model on IMDB, and Table 6 shows the result of attacking ERNIE model on Waimai.. We can see that evaluate module significantly improves the detection accuracy both on clean data and texts with non-adv spelling errors. Meanwhile , it only slightly decreases the accuracy of adversarial detection. The detection method combined with detector and evaluator is suitable for both English and Chinese data.

Table 5. The Ablation Study of English Data

Method	Clean Acc	SpeErr Acc	English Acc		
			Textfooler	HLBB	PSO
Dec	80.3	75.1	90.2	91	86.9
Dec+Eva	98.7	97.8	88.5	89.6	85.4

Table 6. The Ablation Study of Chinese Data

Method	Clean Acc)	SpeErr Acc	Chinese Acc		
			CWC	CWA	WordHanding
Dec	78.4	70.1	88.5	89.1	91
Dec+Eva	97.2	95.8	86.7	87.2	89.5

5 Conclusion

Facing the vulnerability of LLMs to adversarial attacks, in this paper, we propose a detection method against word-level adversarial samples based on the distribution of word importance. Initially, potential adversarial samples are identified swiftly through a detection module. Then the evaluation module attempts to restore the correct text through word replacement and evaluate their adversary to the victim model. Finally, the non-adversarial samples are filtered out. Experimental results demonstrate that our method achieves higher detection accuracy on both clean texts and adversarial samples compared to baseline. In the future, we will further explore detection methods for a broader range of adversarial attacks.

6 Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62172051 and 62272052).

References

1. Jiang, W., He, Z., Zhan, J., Pan, W., & Adhikari, D. (2021). Research progress and challenges on application-driven adversarial examples: A survey. ACM Transactions on Cyber-Physical Systems (TCPS), 5(4), 1-25.
2. Wang, W., Wang, R., Wang, L., Wang, Z., & Ye, A. (2021). Towards a robust deep neural network against adversarial texts: A survey. ieee transactions on knowledge and data engineering, 35(3), 3159-3179.
3. Zou, A., Wang, Z., Kolter, J. Z., & Fredrikson, M. (2023). Universal and transferable adversarial attacks on aligned language models. arXiv preprint arXiv:2307.15043.
4. Zhu, K., Wang, J., Zhou, J., Wang, Z., Chen, H., Wang, Y., ... & Xie, X. (2023). Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. arXiv preprint arXiv:2306.04528.
5. Pruthi, D., Dhingra, B., & Lipton, Z. C. (2019, July). Combating Adversarial Misspellings with Robust Word Recognition. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 5582-5591).
6. Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2020, April). Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 05, pp. 8018-8025).
7. Su, H., Shi, W., Shen, X., Xiao, Z., Ji, T., Fang, J., & Zhou, J. (2022, May). Rocbert: Robust chinese bert with multimodal contrastive pretraining. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 921-931).
8. Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. Computer Science Review, 40, 100379.
9. Li, L., Ma, R., Guo, Q., Xue, X., & Qiu, X. (2020, November). BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 6193-6202).

10. Garg, S., & Ramakrishnan, G. (2020, November). BAE: BERT-based Adversarial Examples for Text Classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 6174-6181).
11. Maheshwary, R., Maheshwary, S., & Pudi, V. (2021, May). Generating natural language attacks in a hard label black box setting. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 15, pp. 13525-13533).
12. L. XG., L. H, and S. Y, "Adversarial sample generation method based on Chinese features," Journal of Software, vol. 34, no. 11, p. 5143, 11 2023.
13. W. WQ., W. R, W. LN, and T. BX, "Adversarial examples generation approach for tendency classification on Chinese texts," Journal of Software, vol. 30, no. 8, p.2415, 2019.
14. T. X., W. LN, W. RZ, and W. JY, "A generation method of word-level adversarial samples for chinese text classification," Netinfo Security, no. 9, pp. 12–16, 2020.
15. Jia, R., & Liang, P. (2017, September). Adversarial Examples for Evaluating Reading Comprehension Systems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 2021-2031).
16. Pruthi, D., Dhingra, B., & Lipton, Z. C. (2019, July). Combating Adversarial Misspellings with Robust Word Recognition. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 5582-5591).
17. Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.J., Srivastava, M., Chang, K.W.: Generating natural language adversarial examples. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2890-2896 (2018)
18. Yoo, J. Y., & Qi, Y. (2021, November). Towards Improving Adversarial Training of NLP Models. In Findings of the Association for Computational Linguistics: EMNLP 2021 (pp. 945-956).
19. Jones, E., Jia, R., Raghunathan, A., & Liang, P. (2020, July). Robust Encodings: A Framework for Combating Adversarial Typos. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 2752-2765).
20. Sun, Z., Li, X., Sun, X., Meng, Y., Ao, X., He, Q., Wu, F., Li, J.: Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 20652075 (2021)
21. Alon, G., & Kamfonas, M. (2023). Detecting language model attacks with perplexity. arXiv preprint arXiv:2308.14132.
22. Zhou, Y., Jiang, J. Y., Chang, K. W., & Wang, W. (2019, November). Learning to Discriminate Perturbations for Blocking Adversarial Attacks in Text Classification. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 4904-4913).
23. Wang, X., Yifeng, X., & He, K. (2022, August). Detecting textual adversarial examples through randomized substitution and vote. In Uncertainty in Artificial Intelligence (pp. 2056-2065). PMLR.
24. Zang, Y., Qi, F., Yang, C., Liu, Z., Zhang, M., Liu, Q., & Sun, M. (2020, July). Word-level Textual Adversarial Attacking as Combinatorial Optimization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 6066-6080).
25. Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies (pp. 142-150).

26. X. Zhang., J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” Advances in neural information processing systems, vol. 28, 2015.
27. Pang, B., & Lee, L. (2005, June). Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (pp. 115-124).
28. Du, Z., Qian, Y., Liu, X., Ding, M., Qiu, J., Yang, Z., Tang, J.: Glm: General language model pretraining with autoregressive blank infilling. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 320-335 (2022)