Adversarial Examples Generation Method for Chinese Text Classification

En-Hui Xu[1], Xiao-Lin Zhang[1], Yong-Ping Wang[1], Shuai Zhang[1], Li-Xin Liu[2], and Li Xu[3]
(Corresponding author: Xiao-Lin Zhang)

School of Information Engineering, Inner Mongolia University of Science and Technology[1]
Baotou 014010, China
School of Information, Renmin University of China[2]
Beijing 100872, China
Baotou Medical College, Inner Mongolia University of Science and Technology[3]
Baotou 014010, China
(Email: zhangxl@imust.edu.cn)
(Received Nov. 22, 2021; Revised and Accepted Apr. 2, 2022; First Online Apr. 10, 2022)

Abstract

Aiming at the problem that DNNs-based text classification systems are vulnerable to adversarial example attacks, a method of adversarial example generation for Chinese text classification, WordHit, is proposed. In this method, we use the morphological and phonological features of Chinese characters to establish a pool of similar characters and homophones, find important words or phrases that affect classification by removing non-contributing clauses and calculating word importance scores and design a modification strategy that combines word sound and word form to generate adversarial examples to achieve a black-box attack on Chinese text classification models. The word-CNN model and the BiLSTM model are used to verify the effectiveness and versatility of different classification tasks. It is proved that the adversarial example generated by this method can be effectively transferred to the BERT model and the actual deployed sentiment analysis system.

Keywords: Adversarial Examples; Black-box Attack; Chinese Character Characteristics; Chinese Text Classification; Deep Neural Networks

1 Introduction

In recent years, Deep Neural Networks (DNNs) [19] have been widely used in many fields such as computer vision, speech recognition and natural language processing, however, Szegedy *et al.* [15] found that these neural network models are exceptionally vulnerable to adversarial example attacks. To address the security problem, many adversarial example generation methods such as FGSM (Fast Gradient Sign Method) [5], Deepfool [10], C&W [1], and PGD (Project Gradient Descent) [9] have been proposed one after another. However, most of these methods are targeted at images [17], and the discrete properties of text and metrics different from those of images make these methods not directly applicable to text. In the textual domain, Papernot *et al.* [11] first proposed to generate adversarial examples, where the authors used FGSM to find adversarial perturbations to modify the word vector, however, the modified word vector may not have words corresponding to it, so the authors used a specific dictionary to select words to replace the original words. Although the mapping problem is solved, more words unrelated to the original word are introduced, resulting in grammatical errors. In addition, Samanta *et al.* [14] used FGSM to locate important words and created a candidate pool for each word in advance, and then modified the first k important words using three strategies: insertion, replacement and deletion. However, there may be some important words without candidates in the actual input. The above two methods are performed in a white-box scenario [20], and Gao *et al.* [3] studied the adversarial example generation method in a black-box scenario [4, 6, 7] and proposed the DeepWordBug algorithm. The algorithm uses the output of the model to find the keywords in the original text by the word importance calculation function, and then generates adversarial examples using insertion, deletion, replacement and exchange of characters. Ren *et al.* [13] proposed a greedy algorithm for word-level attacks by first determining the keywords to be replaced by Probability Weighted Word Saliency (PWWS) to determine the order of keywords to be replaced, and then use WordNet to find synonyms to generate adversarial examples. Similar to Ren *et al.* Zang *et al.* [18] proposed a word-level attack algorithm based on sememes, which can generate more diverse adversarial examples by finding the words with the same sememes corresponding to each word in the original sample through HowNet and then using a particle swarm

algorithm to optimize the combination of candidate words in the discrete space.

The above adversarial example generation methods are designed based on English text and have not been studied for Chinese text, and certain character-level modification strategies are not applicable to Chinese. Wang *et al.* [16] first proposed an adversarial example generation method for Chinese, and they used a pre-trained substitution model and word importance calculation function to determine the keywords to be replaced, and used homophones for replacement. Since the method only uses the phonetic features of Chinese characters for keyword replacement, the features are not fully utilized and the modification strategy is relatively single. Based on the above research work, we propose an adversarial example generation method WordHit for Chinese text, which constructs a candidate pool by analyzing the word form and phonetic features of Chinese characters, then calculates the important words affecting the model classification by a new screening algorithm of important words or phrases, and finally modifies the important words using a modification strategy that combines word sound and word form to generate adversarial examples. A word-level black box attack against Chinese text under a multi-scene classification task is effectively implemented. The main contributions of this paper are follows.

1) WordHit, an adversarial example generation method for Chinese text, is proposed to generate adversarial examples by only slightly modifying the original text without the need to understand the target model parameters. It can interfere with classification tasks of multiple scenarios, such as sentiment analysis, spam classification and news classification.

2) Candidate pools of similar characters and homophones are established for Chinese texts, and the candidate words maintain high semantic similarity with the original words, which can effectively ensure the quality and diversity of the generated adversarial examples.

3) A new filtering method for important words or phrases is designed, which can effectively identify the key words affecting the model decisions under different classification tasks, reduce the modification rate, and generate adversarial examples at a smaller cost.

4) Experiments on real datasets using WordHit to attack word-CNN and BiLSTM in the sentiment analysis task reduce the model accuracy to below 50%, and the attack effectiveness is better than baseline methods. In the spam classification task and news classification task, it also reduces the model accuracy to around 50%, demonstrating the generality of the adversarial example generation algorithm. In addition, the transferability of adversarial examples is successfully exploited to attack BERT and two actually deployed sentiment analysis systems.

The rest of this article is organized as follows. The text adversarial example is described and defined in detail in Section 2. The adversarial example generation method WordHit is introduced in Section 3. In Section 4, experiments are conducted on four real datasets. Finally, the conclusion of the paper is given in Section 5.

2 Text Adversarial Examples

Given a data set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ with n texts and a set of corresponding n labels $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$. A pre-trained natural language classification model F, which needs to learn the mapping $f : \mathcal{X} \to \mathcal{Y}$ from the input text $x \in \mathcal{X}$ to the label $y \in \mathcal{Y}$. Finally, it can classify the original input text x to the true label y_{true} as much as possible, as shown in Equation (1):

$$\arg\max_{y_i \in \mathcal{Y}} P\left(y_i \mid x\right) = y_{\text{true}} \,. \tag{1}$$

Under normal circumstances, an adversarial example x' is generated by adding a small disturbance r to x. The adversarial example will cause model F to give a wrong label, as in Equation (2):

$$\arg\max_{y_i \in \mathcal{Y}} P\left(y_i \mid x'\right) \neq y_{\text{true}} \,. \tag{2}$$

At the same time, the disturbance is required to be imperceptible to the human eye ,which means it will not cause significant changes in semantics, ensuring that humans can still understand the meaning of the original text. So the adversarial example can be defined as in Equation (3):

$$\begin{gathered} x' = x + r, \quad \|r\|_p < \epsilon, \\ \arg\max_{y_i \in \mathcal{Y}} P\left(y_i \mid x'\right) \neq \arg\max_{y_i \in \mathcal{Y}} P\left(y_i \mid x\right). \end{gathered} \tag{3}$$

In Equation (3), $\|r\|_p$ defined in Equation (4) uses p-norm to represent the constraint on perturbation r, and L_0, L_2 and L_∞ are commonly used.

$$\|r\|_p = \left(\sum_{i=1}^{n} |w_i^* - w_i|^p\right)^{\frac{1}{p}}. \tag{4}$$

In Equation (4), the original input text is expressed as $x = w_1 w_2 \ldots w_i \ldots w_n$, where $w_i \in \mathbb{D}$ is a word and $\mathbb{D}$ is a dictionary of words. In order to make the perturbation small enough to be undetectable by humans, the textual adversarial examples need to satisfy word constraints, grammar constraints, and semantic constraints. The word constraint requires that the modified word cannot be the wrong word, the grammar constraint is to ensure the grammatical correctness of the adversarial example, and the semantic constraint ensures that the generated adversarial example should retain the original semantic information. To satisfy the above constraints, homophones and similar characters are used to achieve modification of Chinese text, and a maximum modification threshold σ is

set to constrain the modification magnitude of the adversarial examples. Thus, the effective adversarial example x' can be further expressed Equation (5).

$$F(x') \neq F(x), \text{Cost}(x', x) \leq \sigma. \quad (5)$$

where Cost $(\cdot)$ is the cumulative frequency of text modification.

3 WordHit

DNNs-based text classification system classifies text based on its features, not every word plays the same role in the classification label, some key words closely affect the classification result, and changing key words can largely change the original classification label. In this paper, we visit the target model to locate the keywords that affect the classification. The specific process is shown in Figure 1.

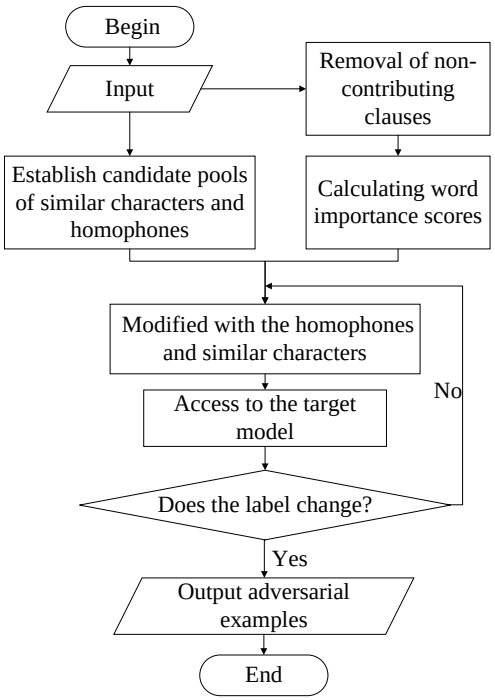


Figure 1: Adversarial example generation process

The process is described as follows.

① Establish candidate pools of similar characters and homophones: analyze the character shape and phonetic characteristics of Chinese text, and establish a candidate pool of similar characters and homophone candidates for each Chinese character.

② Removal of non-contributing clauses: The original text is divided into clauses, and the result of removing each clause in turn is input to the target model to obtain a score relative to the correct label. By calculating the difference with the original score, the clauses that do not contribute to the current classification label are removed.

③ Calculating word importance scores: The retained clauses are divided into words and filtered out stop words, each word is marked as UNK in turn, and then the marked results are input to the target model, and the important words or phrases that affect the classification of the model are ranked by calculating word importance scores.

④ Modified with the homophones and similar characters: one important word or phrase is selected at a time in the order of ranking, and it is modified into the corresponding word in the candidate pool using the word sound and word form modification strategy.

The modified results are input to the target model to obtain the classification labels, and if the labels do not change, the execution continues ④ until the predicted labels of the target model change or reach the modification threshold, and the final generated adversarial examples are output.

3.1 Establish Candidate Pools of Similar Characters and Homophones

The word level-based text adversarial example generation method usually requires a candidate pool, and the quality of the candidate pool determines the quality of the adversarial example to a certain extent. The degree of similarity between Chinese characters can be measured by the sameness or similarity of "sounds", or the similarity of "shapes". Taking the glyph and phonetic features of Chinese characters as the analysis object, a candidate pool is generated for each Chinese character.

3.1.1 Candidate Pool of Similar Characters

In order to analyze the similarity relationship between characters more flexibly, a simple character similarity comparison method is designed. Commonly used Chinese characters are obtained from GB1312 area code, and under a given font, character bitmaps are obtained and rendered into fixed-size pictures with grayscale values between 0 and 255 for each pixel point. Each picture is converted into a vector according to the size of the grayscale value, and the degree of similarity between Chinese characters is analyzed by comparing the Euclidean distance between two vectors. Following this strategy, for each Chinese character, the top n characters closest to it can be found. Table 1 shows the morphological similarities corresponding to some Chinese characters when n is taken as 1,2,3.

In order to maximize the perceptual similarity between the generated candidate pool of similar character and the original character, the optimal candidate for each character is obtained by taking $n = 1$ and adding it to the candidate pool of similar character. Considering that the

Table 1: Candidate Pool of Similar Characters and Homophonic Characters Corresponding to Some Chinese Characters. Ori.(Original words). BS(The best similar characters). TC(Traditional Chinese). SW(Splitting words).CPoSC(Candidate pool of similar characters). CPoH(Candidate pool of homophones)

Ori.	Similar words			CPoSC			CPoH		
	$n=1$	$n=2$	$n=3$	BS	TC	SW	Homophones of similar shape	Pinyin	Other homophones
评	怦	坪	抨	怦	評	讠平	坪枰呯玶蚜胖鲆鲆平苹萍	Ping	凭屏瓶
假	徦	瑕	蝦	瑕	/	亻叚	徦婽椵叚瘕	Jia	甲钾胛贾玾价加家驾
睬	睐	踩	眯	睐	/	目采	踩婇採綵綵埋啋采菜彩	Cai	蔡材财猜
惠	恵	蕙	崽	恵	/	/	恵蕙澮憓德橞璓蟪	Hui	会回汇荟绘慧
鲜	鲆	鲑	蚌	鲆	鮮	鱼羊	鲜薢灕薢廯	Xian	先仙闲显险掀现限线
真	直	具	其	直	/	/	禛媜眞逭	Zhen	针阵珍贞甄臻祯桢侦镇

simplified and traditional forms of a Chinese character can represent the same semantic information, the corresponding traditional form of the character is added to the candidate pool. The splitting of a left-right structure does not affect human reading, so if the character has a left-right structure, the split character is added to the candidate pool as a candidate. The results of the morphological candidate pool are shown in Table 1.

3.1.2 Candidate Pool of Homophones

Unlike similar characters, a Chinese character often has many homophones. The homophones are further divided into two parts: homophones of similar shape and other homophones. The homophones of similar shape can have a certain degree of morphological similarity while ensuring similar pronunciation, while the other homophones only require similar pronunciation. If the candidate pool does not reach the set capacity, the pinyin and other homophones of the word are added to the candidate pool. In the experiment, the maximum capacity of the homophone candidate pool was set to 15, and n was set to 20. The results of the homophone candidate pool are shown in Table 1.

3.2 Important Words or Phrases Filtering Algorithm

To ensure the readability and validity of the text obtained after modification, modifying important words or phrases in the text and controlling the magnitude of the changes are the basic strategies of text adversarial generation. Which are the important words and how to locate them are the problems to be solved by this algorithm.

3.2.1 Remove Non-contributing Clauses

An input sample often contains multiple sentences, but not every sentence contributes to the classification label. In order to locate key words more efficiently, meaningless sentences need to be eliminated. First, the original sample x is divided into n clauses using punctuation marks to obtain $x = \{s_1, s_2, \ldots, s_n\}$. For the i-th clause in the sequence, the confidence difference (Delete Score, DS) between the input after removing the clause and the original input is calculated in turn as shown in Equation (6).

$$DS(s_i) = F(s_1, \ldots, s_{i-1}, s_i, s_{i+1}, \ldots, s_n) - F(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n) \tag{6}$$

If $DS(s_i) \leq 0$, then it means that s_i does not contribute to the current classification label of the sequence and it is removed from the original input. After n visits to the target model, the final filtered sequence X is obtained.

3.2.2 Calculating Word Importance Scores

The sequence X obtained by the first filtering operation contains all the clauses that contribute to the current classification label. Then, the important words or phrases that affect the classification in all clauses are found by calculating the word importance scores. All the sentences in X are divided and the meaningless stop words are filtered out to obtain: $X = \{w_1, w_2, \ldots, w_i, \ldots, w_n\}$. After marking each word in turn as an unknown character UNK, it is input to the target model, and the difference between the original score returned by the target model and the current score is counted as the importance score of the word, as shown in Equation (7).

$$S(w_i) = F(w, \ldots, w_{i-1}, w_i, w_{i+1}, \ldots, w_n) - F(w_1, \ldots, w_{i-1}, \text{UNK}, w_{i+1}, \ldots, w_n) \tag{7}$$

Finally, each word is sorted by importance score from highest to lowest to get the list of words to be modified X'.

3.3 Modified with the Homophones and Similar Characters

Combining the established candidate pool with the list of words to be modified X', a modification strategy $\boldsymbol{T}$ that combines word sound and word form is proposed. The specific description is as follows.

(1) Select a word or phrase to be modified in the vocabulary list X' in order.

(2) The corresponding candidate pool of similar characters and candidate pool of homophones are selected with probabilities of k $(k \in [\ 0,\ \ 1\ ])$ and $1-k$, respectively.

(3) If the pool selected is the candidate pool of similar characters, the candidate words in the pool are randomly selected for replacement; If the pool selected is the candidate pool of homophones, the homophones of similar shape is first selected with probability j $(j \in [\ 0,\ \ 1\ ])$, and the pinyin and other homophones are selected with probability $1-j$, and then the words in the selected range are randomly selected for replacement.

The complete WordHit algorithm is shown in Algorithm (1).

Algorithm 1 WordHit

Input: Text x; Target classification model F; Modify strategy $\boldsymbol{T}$; Modify threshold δ
Output: Adversarial example x'

Begin
$x = \{s_1, s_2, \ldots, s_n\}$
for $i = 1, \cdots, n$ **do**
  $DS(s_i) = F(s_1, \ldots, s_{i-1}, s_i, s_{i+1}, \ldots, s_n) - F(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$
  if $DS(s_i) \leq 0$ **then**
    $x \leftarrow$ Delete s_i from x
  end if
end for
$X \leftarrow x$
$X = \{w_1, w_2, \ldots, w_i, \ldots, w_n\}$
for i = 1, $\cdots$, n **do**
  $S(w_i) = F(w, \ldots, w_{i-1}, w_i, w_{i+1}, \ldots, w_n) - F(w_1, \ldots, w_{i-1}, \mathrm{UNK}, w_{i+1}, \ldots, w_n)$
end for
$X' \leftarrow$ sort w_i by descending $S(w_i)$
for w_i in X' **do**
  $w_i^* = \boldsymbol{T}(w_i)$
  $x' \leftarrow$replace w_i with w_i^* in x'
  if $F(x') \neq F(x)$ and Cost $(x', x) \leq \sigma$ **then**
    return x'
  end if
end for

4 Empirical Evaluation

4.1 Experiment Setting

Four publicly available datasets were used for validation and performance evaluation of the WordHit algorithm, and the specific data information is shown in Table 2. word-based CNN (word-CNN) [12] and Bi-directional LSTM (BiLSTM) [8] were used as the target models for the experiments. Among them, the word-CNN consists of a 300-dimensional embedding layer, three convolutional layers and a fully connected layer, and the convolutional layer consists of 256 convolutional kernels of size 2,3,4 with a step size of 1. The BiLSTM consists of a 300-dimensional embedding layer, a bidirectional LSTM layer and a fully connected layer, and the forward and backward directions of the bidirectional LSTM layer consist of 64 LSTM units, respectively. The forward and backward directions of the bi-directional LSTM layer are composed of 64 LSTM cells respectively.

4.2 Comparison of Experimental Methods

Validation of the effectiveness of the WordHit algorithm on the sentiment analysis task and comparison of two baseline algorithms: DeepWordBug [6] and WordHanding [16]. The performance of the WordHit algorithm was evaluated separately on the spam classification task and the news classification task. In the experiments, the maximum modification threshold δ was set to 11 for the news classification task and 30 for the other tasks, and WordHit selected the strategy of phonological modification for important words or phrases, setting k to 0.5 and j to 0.6. The experimental results are shown in Table 3 and Table 4.

For both datasets of the sentiment analysis task, the model accuracy reduction exceeds that of the baseline method, indicating that WordHit outperforms DeepWordBug and WordHanding in misleading classifiers with better attacks. As shown by the experimental results of the spam classification task and the news classification task, the WordHit algorithm succeeds in both classification tasks, bringing down the classification accuracy to about 50%, proving its generality under multi-scene classification tasks.

In order to verify the relationship between the accuracy of model detection and the modification threshold, the same experimental setup as the WordHanding algorithm is maintained, and 1000 samples with length greater than 120 words are selected from the two sentiment analysis datasets respectively, and the adversarial examples are generated by adjusting different modification thresholds to investigate the effect of different thresholds on the effectiveness of the generated adversarial examples. Figure 2 and Figure 3 show the variation curves of the detection accuracy of the two models with the modification threshold on the Ctrip dataset and the JingDong(JD) dataset, respectively. The accuracy of the model detection gradually decreases as the modification threshold increases, i.e., the text has enough modification space to modify the keywords affecting the classification. Compared with the baseline method, the WordHit algorithm can reduce the model detection accuracy significantly by modifying only a few words, and the attack effect has leveled off when the number of modified words reaches 15, indicating that WordHit can minimize the modification of the original sample and ensure the text readability.

Table 2: Statistics on the datasets

Dataset	Classes	Train samples	Test samples	Average words	Task
Ctrip	2	12000	3000	132	Sentiment analysis
JD	2	35000	5000	36	Sentiment analysis
Spam	2	90000	10000	51	Spam classification
THUCNews	10	90000	10000	19	News classification

Table 3: Validation of the algorithm WordHit on sentiment analysis tasks

(a) word-CNN

Dataset	Ori_acc(%)	Baselines				Ours	
		DeepWordBug		WordHanding		WordHit	
		Accuracy(%)	Reduction	Accuracy(%)	Reduction	Accuracy(%)	Reduction
Ctrip	92.03	77.48	14.55	69.53	22.50	41.07	50.96
JD	91.46	75.04	16.42	70.43	21.03	48.42	43.04

(b) BiLSTM

Dataset	Ori_acc(%)	Baselines				Ours	
		DeepWordBug		WordHanding		WordHit	
		Accuracy(%)	Reduction	Accuracy(%)	Reduction	Accuracy(%)	Reduction
Ctrip	92.03	77.48	14.55	69.53	22.50	41.07	50.96
JD	91.46	75.04	16.42	70.43	21.03	48.42	43.04

4.3 Adversarial Examples Quality Measurement

To measure the quality of the generated adversarial examples, the WMD (Word Mover's Distance) is used to measure the similarity between adversarial examples and the original samples. The smaller the WMD, the higher the semantic similarity. 1000 data and their corresponding adversarial examples are randomly selected from two datasets for the experiment. The proportion of WMDs in different intervals is shown in Figure 4.

WordHit algorithm has the largest percentage on the interval of 0-0.2, which indicates that the adversarial examples generated by WordHit algorithm have higher semantic similarity with the original samples. The WMD of the adversarial examples generated by WordHit algorithm in the interval of 0-0.6 accounts for 78.3% of the data, which is higher than the baseline method, indicating that most of the generated adversarial examples have higher quality and can better retain semantic information. Table 5 shows examples of the adversarial examples generated using the WordHit algorithm, which can be seen that the generated adversarial examples retain the original semantics and can be understood by humans.

4.4 Human Evaluation

To further explore the impact of the adversarial examples generated by the WordHit algorithm on human reading, we conducted a human evaluation. The main two aspects of the evaluation were to assess the accuracy of human classification of the generated adversarial examples and to assess the naturalness of the adversarial examples from the perspective of human perception. This was done by randomly selecting 100 clean and corresponding confrontation samples from the two sentiment analysis datasets, disrupting the samples and giving them to volunteers for classification, and giving them a likelihood score between 1 and 5, the higher the score, the more likely the human was to write the article. A total of six volunteers participated in the experiment, and Table 6 shows the evaluation results.

As can be seen from Table 6, although adversarial examples make the model misclassify, the human classification effect is still very good, and the difference between the classification accuracy and that of the clean samples is less than 5%, indicating that the adversarial examples generated by the WordHit algorithm retain the semantic information of the original text better and do not affect people's reading comprehension of the text content. In terms of the naturalness score, although it is lower than that of the clean sample, the difference is small, indicating that the naturalness of the adversarial example is within the range acceptable to humans.

4.5 Transferability Assessment

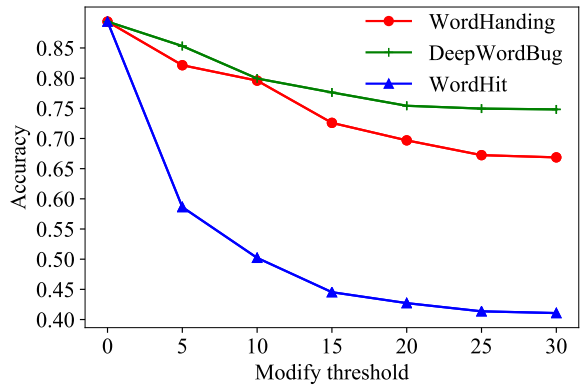
In the field of text classification, the transferability of adversarial examples means that the adversarial examples generated against one classification model can be used to attack other models as well. Using the transferability of adversarial examples, adversarial examples can be gener-

Table 4: Evaluating WordHit performance on spam classification and news classification tasks

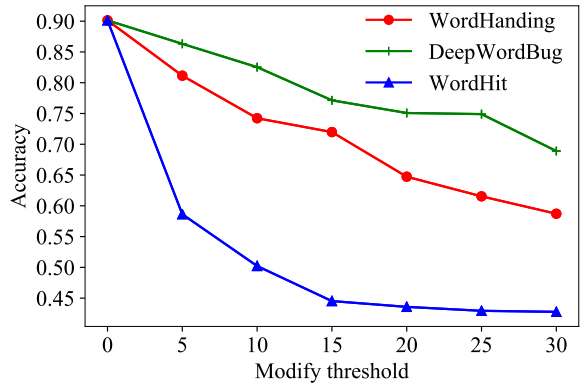
Dataset	Model	Ori_acc(%)	WordHit	
			Accuracy(%)	Reduction
Spam	word-CNN	99.86	49.95	49.91
	BiLSTM	99.76	49.78	49.98
THUCNews	word-CNN	90.89	53.40	37.49
	BiLSTM	90.64	52.79	37.85

Table 5: Examples of original samples and generated adversarial examples

Original sample	Label	Adversarial example	Label
房间巨小, 电视成了摆设, 开不了, 服务员态度冷漠, 不睬我, 以后不会再进这家酒店.	Negative	房间剧 xiao, 电视成了摆设, 开丕了, 服务员态度泠莫. bu 睐我, 以后布绘再进这家酒店.	Positive
东西非常实惠, 快递真给力, 昨天下的单今天就到了, 新包装颜色鲜艳, 好评.	Positive	东西非常湿蕙, 快递直铪厉, 昨天下的单今天就到了, 新包装颜色鲜燕,女子評.	Negative
用完特别容易痒, 跟以前屈臣氏的根本不一样, 绝对的假货! 贪小便宜吃大亏!	Negative	用完特别容易养, 跟以前屈臣氏的裉奔步一样, 绝对的嘏货!谈小便宜吃大兮!	Positive
酒店脚摩很有特色, 住客还给打折, 房间装修尚可, 位置稍偏.	Positive	酒店脚摩恨油牛 寺涩, 住客还給 Da 浙, 房间装修伤渴, 位置稍偏.	Negative

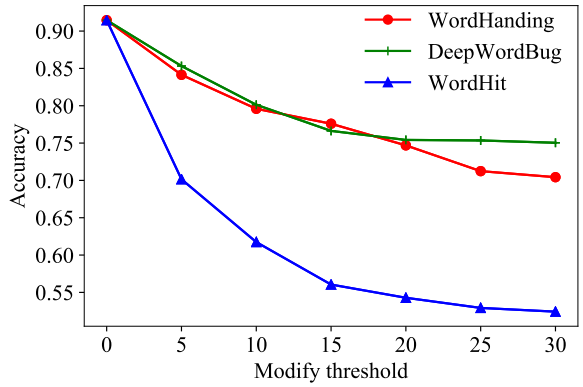


(a) word-CNN

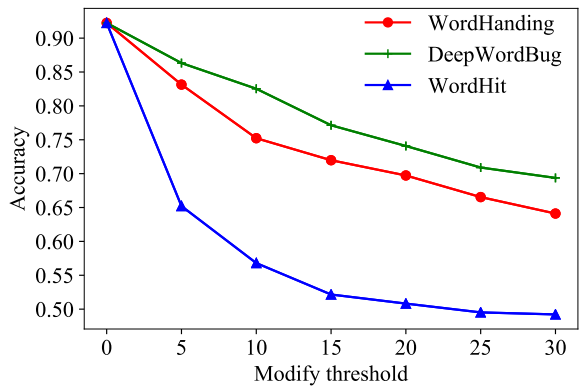


(b) BiLSTM

Figure 2: The variation curve of detection accuracy with modification threshold for the adversarial example of Ctrip review dataset



(a) word-CNN



(b) BiLSTM

Figure 3: The variation curve of detection accuracy with modification threshold for the adversarial example of JD review dataset

Table 6: Comparison with human evaluation.

Dataset	Model	Examples	Accuracy of model(%)	Accuracy of human(%)	Score[1-5]
Ctrip	word-CNN	Original	97.00	97.00	4.70
		Adversarial	18.00	93.00	3.83
	BiLSTM	Original	93.00	97.00	4.70
		Adversarial	21.00	93.00	3.83
JD	word-CNN	Original	96.00	98.00	4.50
		Adversarial	23.00	95.00	3.95
	BiLSTM	Original	95.00	98.00	4.50
		Adversarial	25.00	95.00	3.95

Table 7: Results of adversarial examples generated using BiLSTM model to attack other models/systems

Dataset	Model/Cloud Platform	Ori_acc(%)	WordHit	
			Accuracy(%)	Reduction
Ctrip	word-CNN	92.03	57.37	34.66
	BERT	91.60	58.40	33.20
	Tencent Cloud	87.67	61.40	26.27
	Baidu AI	88.43	63.10	25.33
JD	word-CNN	91.46	61.37	30.09
	BERT	92.52	62.78	29.74
	Tencent Cloud	89.42	65.44	23.98
	Baidu AI	88.90	63.78	25.12

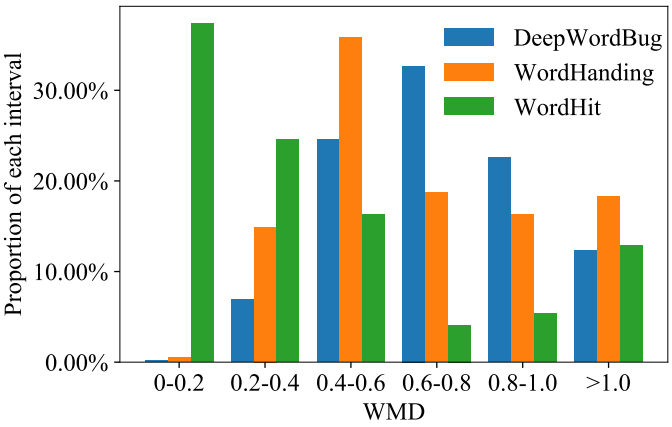


Figure 4: Proportion of the number of samples in different WMD distance intervals to the total samples

ated on alternative models to achieve a black-box attack on the target model. To evaluate the transferability of the adversarial examples generated by the WordHit method, in addition to the word-CNN model and the BiLSTM model, the BERT [2] model, and two actual deployed sentiment analysis systems (Tencent Cloud Sentiment Analysis System and Baidu Cloud Sentiment Analysis System) were additionally introduced. Table 7 and Table 8 show the results of adversarial example attacks on other models (systems) generated using the BiLSTM model and the word-CNN model, respectively. The results show that the adversarial examples generated by the WordHit algorithm for both models can be effectively transferred to the other four models (systems), causing the classification accuracy of the word-CNN, BILSTM and BERT models to drop by about 30% and the accuracy of the two sentiment analysis systems to drop by about 25%.

4.6 Adversarial Training

Adversarial training is a technique to improve model robustness by adding adversarial examples to the training set and repeating the training to improve model robustness, which is commonly used in image classification and can also be used as a means to enhance model generalization in natural language processing tasks. To analyze the effect of adversarial training on classification accuracy, 5000 data items are randomly selected from the Ctrip hotel review dataset, and adversarial examples are generated as the ensemble $\mathbb{A}$ using WordHit on the BiLSTM model. Several adversarial examples are randomly selected from the ensemble $\mathbb{A}$ and added to the original training set to evaluate the classification accuracy of the original test set and the classification accuracy of the adversarial example test set.

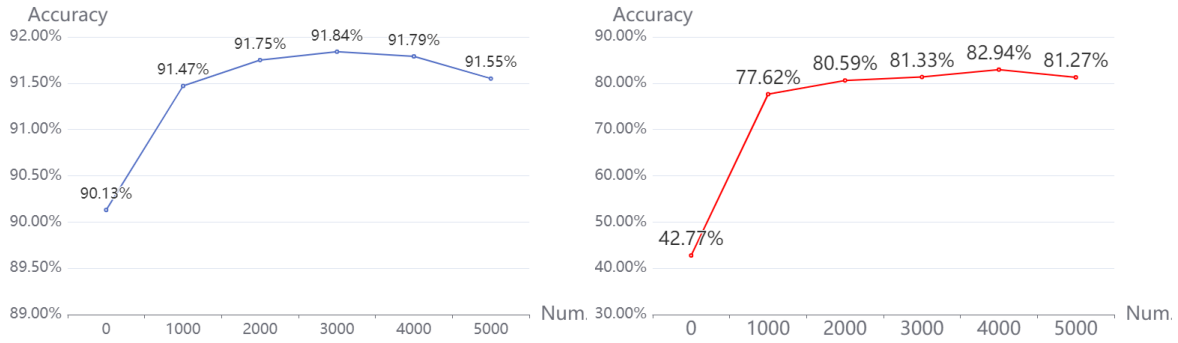
The data in Figure 5(a) show that the adversarial training helps to improve the accuracy of the classification model. Figure 5(b) illustrates that when more and more adversarial examples are involved in the training, the robustness of the model steadily improves and the classification accuracy can be improved to over 80%.

5 Conclusions

Adversarial example generation for Chinese text classification models is important for evaluating and improving

Table 8: Results of adversarial examples generated using word-CNN model to attack other models/systems

Dataset	Model/Cloud Platform	Ori_acc(%)	WordHit	
			Accuracy(%)	Reduction
Ctrip	BiLSTM	90.13	54.33	35.80
	BERT	91.60	60.57	31.03
	Tencent Cloud	87.67	59.83	27.84
	Baidu AI	88.43	62.50	25.93
JD	BiLSTM	92.24	61.52	30.72
	BERT	92.52	63.48	29.04
	Tencent Cloud	89.42	62.38	27.04
	Baidu AI	88.90	64.22	24.68



(a) Accuracy of original sample test set

(b) Accuracy of adversarial example test set

Figure 5: Effect of adversarial training on the classification accuracy of original and adversarial examples

Chinese text classification systems. In the WordHit algorithm, we use the word form and speech features of Chinese characters to construct a candidate pool, find the keywords or phrases affecting classification by important sentence screening and word importance calculation, and design a modification strategy for adversarial example generation, finally realizing a word-level black box attack against Chinese text classification models under a multi-scene classification task. The experimental results show that the adversarial examples generated by the WordHit algorithm effectively reduce the classification accuracy while retaining the semantic information of the original text with good readability. The vulnerability of current Chinese text classification models is revealed by attacking the BERT model and the actual deployed sentiment analysis system using the transferability of adversarial examples, and the impact of adversarial training on the classification accuracy of the original test set and the adversarial example test set is further analyzed. Therefore, we use this as the basis for our work on the defense of such attack algorithms in the next phase, and explore more robust deep learning models and methods.

Acknowledgments

This work was partially supported by the Natural Science Foundation of China (No.61562065) and the Natural Science Foundation Project of Inner Mongolia (No.2019MS06001 and No.2019MS06036). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017.

[2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

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

[4] S. Garg and G. Ramakrishnan, "Bae: Bert-based adversarial examples for text classification," *arXiv preprint arXiv:2004.01970*, 2020.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *Computer Science*, 2014.

[6] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "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, 2020, pp. 8018–8025.

[7] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu, "Bert-attack: Adversarial attack against bert using bert," *arXiv preprint arXiv:2004.09984*, 2020.

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

[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017.

[10] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Computer Vision & Pattern Recognition*, 2016.

[11] N. Papernot, P. McDaniel, A. Swami, and R. Harang, "Crafting adversarial input sequences for recurrent neural networks," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 49–54.

[12] A. Rakhlin, "Convolutional neural networks for sentence classification," *GitHub*, 2016.

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

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

[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *Computer Science*, 2013.

[16] W. Wang, R. Wang, L. Wang, and B. Tang, "Adversarial examples generation approach for tendency classification on chinese texts," *Ruan Jian Xue Bao/J. Softw.*, vol. 30, no. 8, pp. 2415–2427, 2019.

[17] Q. Yaguan, L. Hongbo, J. Shouling, Z. Wujie, W. Shuhui, Y. Bensheng, T. Xiangxing, and L. Jingsheng, "Adversarial example generation based on particle swarm optimization," *Journal of Electronics and Information*, vol. 41, no. 7, pp. 1658–1665, 2019.

[18] Y. Zang, F. Qi, C. Yang, Z. Liu, M. Zhang, Q. Liu, and M. Sun, "Word-level textual adversarial attacking as combinatorial optimization," *arXiv preprint arXiv:1910.12196*, 2019.

[19] H. Zhao, Y. K. Chang, and W. J. Wang, "Research on robustness of deep neural networks based data preprocessing techniques," *International Journal of Network Security*, vol. 24, no. 2, pp. 243–252, 2022.

[20] X. Zheng, J. Zeng, Y. Zhou, C. J. Hsieh, M. Cheng, and X. J. Huang, "Evaluating and enhancing the robustness of neural network-based dependency parsing models with adversarial examples," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6600–6610.

Biography

En-Hui Xu is a graduate student in the School of Information Engineering, Inner Mongolia University of Science & Technology. His research interests include machine learning security and natural language processing.

Xiao-lin Zhang received the Ph.D. degree from the Northeastern University of China, Shenyang, in 2006. She is a professor in the School of Information Engineering, Inner Mongolia University of Science & Technology. Her research interests include database theory and machine learning security, cloud computing, and social network privacy protection.

Yong-ping Wang received a Master's degree from the Wuhan University of Technology in 2010. She is a lecturer in the School of Information Engineering, Inner Mongolia University of Science & Technology. Her research interests are mainly in the field of object detection and data privacy protection.

Shuai Zhang is a graduate student in the School of Information Engineering, Inner Mongolia University of Science & Technology. His research interests include machine learning security and computer vision.

Li-xin Liu is a PhD candidate at the Renmin University of China and the Member of China Computer Federation. Her main research interests include privacy protection and blockchain.

Li Xu received a Master's degree from the Southwest University of Science and Technology in 2009. Her research interest is mainly in the area of image processing and data privacy protection.