applied sciences

MDPI

Article

Universal Adversarial Attack via Conditional Sampling for Text Classification

Yu Zhang †, Kun Shao †, Junan Yang * and Hui Liu

Institute of Electronic Countermeasure, National University of Defense Technology, Hefei 230000, China; zhangy96@yeah.net (Y.Z.); shaokun20@nudt.edu.cn (K.S.); liuhui17c@nudt.edu.cn (H.L.)

* Correspondence: yangjunan@ustc.edu.cn

† These authors contributed equally to this work.

Abstract: Despite deep neural networks (DNNs) having achieved impressive performance in various domains, it has been revealed that DNNs are vulnerable in the face of adversarial examples, which are maliciously crafted by adding human-imperceptible perturbations to an original sample to cause the wrong output by the DNNs. Encouraged by numerous researches on adversarial examples for computer vision, there has been growing interest in designing adversarial attacks for Natural Language Processing (NLP) tasks. However, the adversarial attacking for NLP is challenging because text is discrete data and a small perturbation can bring a notable shift to the original input. In this paper, we propose a novel method, based on conditional BERT sampling with multiple standards, for generating universal adversarial perturbations: input-agnostic of words that can be concatenated to any input in order to produce a specific prediction. Our universal adversarial attack can create an appearance closer to natural phrases and yet fool sentiment classifiers when added to benign inputs. Based on automatic detection metrics and human evaluations, the adversarial attack we developed dramatically reduces the accuracy of the model on classification tasks, and the trigger is less easily distinguished from natural text. Experimental results demonstrate that our method crafts more high-quality adversarial examples as compared to baseline methods. Further experiments show that our method has high transferability. Our goal is to prove that adversarial attacks are more difficult to detect than previously thought and enable appropriate defenses.

Keywords: universal adversarial perturbations; conditional BERT sampling; adversarial attacks; sentiment classification; deep neural networks

check for updates

Citation: Zhang, Y.; Shao, K.; Yang, J.; Liu, H. Universal Adversarial Attack via Conditional Sampling for Text Classification. *Appl. Sci.* **2021**, *11*, 9539. https://doi.org/10.3390/app11209539

Academic Editor: Luis Javier Garcia Villalba, Rafael T. de Sousa, Jr., Robson de Oliveira Albuquerque and Ana Lucila Sandoval Orozco

Received: 4 August 2021
Accepted: 12 October 2021
Published: 14 October 2021

1. Introduction

Deep Neural Networks (DNNs) have made great success in various machine learning tasks, such as computer vision , speech recognition and Natural Language Processing (NLP) [1–3]. However, recent studies have discovered that DNNs are vulnerable to adversarial examples not only for computer vision tasks [4] but also for NLP tasks [5]. The adversary can be maliciously crafted by adding a small perturbation into benign inputs but can trigger the target model to misbehave, causing a serious threat to their safe applications. To better deal with the vulnerability and security of DNNs systems, many attack methods have been proposed further to explore the impact of DNN performance in various fields [6–8]. In addition to exposing system vulnerabilities, adversarial attacks are also useful for evaluation and interpretation, that is, to understand the function of the model by discovering the limitations of the model. For example, adversarial-modified input is used to evaluate reading comprehension models [9] and stress test neural machine translation [10]. Therefore, it is necessary to explore these adversarial attack methods because the ultimate goal is to ensure the high reliability and robustness of the neural network.

These attacks are usually generated for specific inputs. Existing research observes that there are attacks that are effective against any input. In input-agnostic word sequences, when connected to any input of the data set, these tokens trigger the model to produce false predictions. The existence of this trigger exposes the greater security risks of the DNN model because the trigger does not need to be regenerated for each input, which greatly reduces the threshold of attack. Moosavi-Dezfooli et al. [11] proved for the first time that there is a perturbation that has nothing to do with the input in the image classification task, which is called Universal Adversarial Perturbation (UAP). Contrary to adversarial perturbation, UAP is data-independent and can be added to any input in order to fool the classifier with high confidence. Wallace et al. [12] and Behjati et al. [13] recently demonstrated a successful universal adversarial attack of the NLP model. In the actual scene, on the one hand, the final reader of the experimental text data is human, so it is a basic requirement to ensure the naturalness of the text; on the other hand, in order to prevent universal adversarial perturbation from being discovered by humans, the naturalness of adversarial perturbation is more important. However, the universal adversarial perturbations generated by their attacks are usually meaningless and irregular text, which can be easily discovered by humans.

In this article, we focus on designing natural triggers using text-generated models. In particular, we use a BERT-based text sampling method, which is to generate some natural language sentences from the model randomly. Our method sets the enforcing word distribution and decision function that meets the general anti-perturbation based on combining the bidirectional Masked Language Model and Gibbs sampling [3]. Finally, it can obtain an effective universal adversarial trigger and maintain the naturalness of the generated text. The experimental results show that the universal adversarial trigger generation method proposed in this paper successfully misleads the most widely used NLP model. We evaluated our method on advanced natural language processing models and popular sentiment analysis data sets, and the experimental results show that we are very effective. For example, when we targeted the Bi-LSTM model, our attack success rate on the positive examples on the SST-2 dataset reached 80.1%. In addition, we also show that our attack text is better than previous methods on three different metrics: average word frequency, fluency under the GPT-2 language model, and errors identified by online grammar checking tools. Furthermore, a study on human judgment shows that up to 78% of scorers believe that our attacks are more natural than the baseline. This shows that adversarial attacks may be more challenging to detect than we previously thought, and we need to develop appropriate defensive measures to protect our NLP model in the long term.

The remainder of this paper is structured as follows. In Section 2, we review the related work and background: Section 2.1 describes deep neural networks, Section 2.2 describes adversarial attacks and their general classification, Sections 2.2.1 and 2.2.2 describe the two ways adversarial example attacks are categorized (by the generation of adversarial examples whether to rely on input data). The problem definition and our proposed scheme are addressed in Section 3. In Section 4, we give the experimental results with analysis. Finally, we summarize the work and propose the future research directions in Section 5.

2. Background and Related Work

2.1. Deep Neural Networks

The deep neural network is a network topology that can use multi-layer non-linear transformation for feature extraction, and utilizes the symmetry of the model to map high-level more abstract representations from low-level features. A DNN model generally consists of an input layer, several hidden layers, and an output layer. Each of them is made up of multiple neurons. Figure 1 shows a commonly used DNN model on text data: long-short term memory (LSTM).

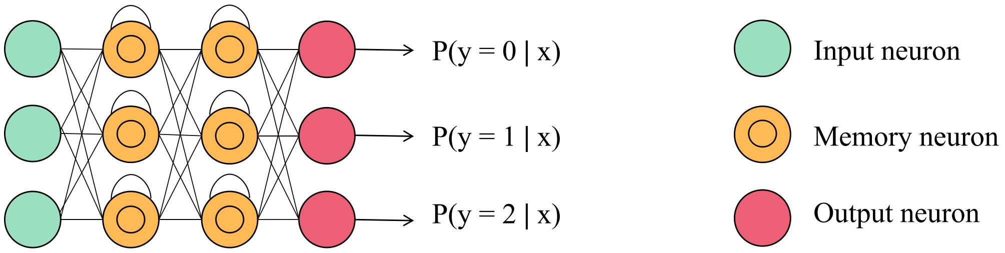

Figure 1. The LSTM models in texts.

The recent rise of large-scale pretraining language models such as BERT [3], GPT-2 [14], RoBertA [15] and XL-Net [16], which are currently popular in NLP. These models first learn from a large corpus without supervision. Then, they can quickly adapt to downstream tasks via supervised fine-tuning, and can achieve state-of-the-art performance on several benchmarks [17,18]. Wang and Cho [19] showed that BERT can also produce high quality, fluent sentences. It inspired our universal trigger generation method, which is an unconditional Gibbs sampling algorithm on a BERT model.

2.2. *Adversarial Attacks*

The purpose of adversarial attacks is to add small perturbations ε in the normal sample x to generate adversarial example x', so that the classification model F makes misclassification. The formula descriptions is $F(x') \neq y$, where $x' = x + \varepsilon$, $|\varepsilon| < \delta$. δ is a threshold to limit the size of perturbations. We classify existing adversarial attack methods according to different criteria. Figure 2 summarizes these categories.

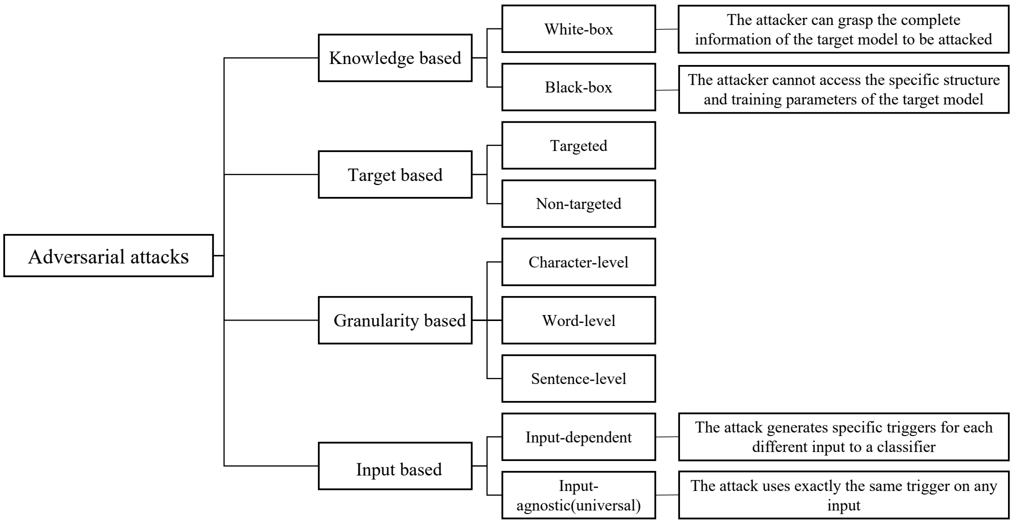
Adversarial attacks
Knowledge based
White-box
The attacker can grasp the complete information of the target model to be attacked
Black-box
The attacker cannot access the specific structure and training parameters of the target model
Target based
Targeted
Non-targeted
Granularity based
Character-level
Word-level
Sentence-level
Input based
Input-dependent
The attack generates specific triggers for each different input to a classifier
Input-agnostic(universal)
The attack uses exactly the same trigger on any input

Figure 2. Categories of adversarial attack methods on textual deep learning models.

According to the attacker's understanding of the model, attacks can be divided into white-box attacks and black-box attacks. In white-box attack, the attack requires the access to the model's full information, including architecture, parametrers, loss functions, activation functions, input and output data. They can obtain excellent adversarial examples. A black-box attack does not require the knowledge about target models, but can access the input and output. This type of attack often relies on heuristics to generate adversarial examples, and it is more practical, as in many real-world applications the details of the DNN is a black box to the attacker.

According to the purpose of adversaries, adversarial attacks can be divided into targeted attacks and non-targeted attacks. In a targeted attack, the generated adversarial example x is deliberately classified into the nth category, which is the target of the attacker.

In a non-directed attack, the adversary is merely to fool the model. The result y' can be any class except for y.

NLP models usually use character encoding or word encoding as model input features, so text adversarial samples can be divided according to the level of disturbance for these features. According to the different attack targets, it can be divided into character-level attacks, word-level attacks, and sentence-level attacks. Character-level attacks act on characters, including letters, special symbols, and numbers. A adversarial sample is constructed by modifying characters in the text, such as English letters or Chinese characters. Different from character-level attacks, the object of word-level attacks is the words in the original input. The primary method is to delete, replace or insert new words in the keywords in the original text. At present, the method of sentence-level attack is to treat the original input of the entire sentence as the object of disturbance, with the intention of generating an adversarial example that has the same semantics as the original input but changes the judgment of the target model. Commonly used sentence level attack methods include paraphrasing, re-decoding after encoding and adding irrelevant sentences.

Whether the generation of adversarial examples depends on each input data, we divide the attack methods into input-dependent adversarial attacks and universal adversarial attacks. Figure 3 shows a schematic diagram of a adversarial attack.

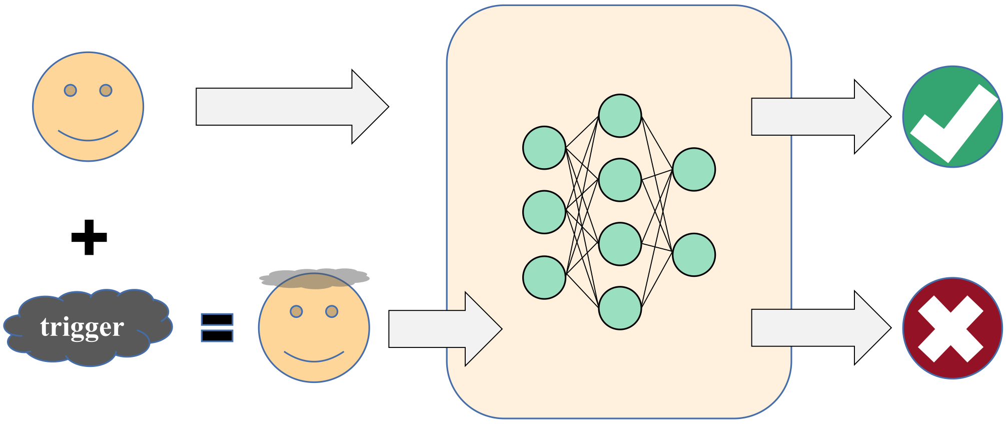

Figure 3. The schematic diagram of adversarial attacks.

2.2.1. Input-Dependent Attacks

These attacks make specific triggers for each different input of the model. Under the white box condition, we can use the model loss function to solve the gradient information and then guide adversarial examples. For example, Papernot et al. [5] disturbed the word embedding vector of the original input text. Ebrahimi et al. [20] carefully designed the character conversion perturbation and used the direction of the model loss gradient to select the best perturbation to replace the words of the benign text, resulting in performance degradation. Lei et al. [21] use embedded transformation to introduce a replacement strategy. Under the black box condition, Alzantot et al. [22] proposed an attack method based on synonym substitution and genetic algorithm. Zang et al. [23] proposed an attack method based on original word replacement and particle swarm optimization algorithm.

2.2.2. Universal Attacks

Wallace et al. [12] and Behjati et al. [13] also proposed a universal adversarial disturbance generation method that can be added to any input text. Both papers used gradient loss to guide the search direction to find the best perturbation to cause as many benign inputs in the data set as possible to fool the target NLP model. However, the attack word sequence generated in these two cases is usually unnatural and meaningless. In contrast, our goal is to obtain a more natural trigger. When a trigger that does not depend on any input samples is added to the normal data, it will cause errors in the DNN model.

3. Universal Adversarial Perturbations

In this section, we are going to formalize the problem of finding the universal adversarial perturbations for a text classifier and introduce our methods.

3.1. Universal Triggers

We seek an input-agnostic perturbation, which can be added to each input sample and deceive a given classifier with a high probability. If the attack is universal, the adversarial threat is higher: use the same attack on any input [11,24]. The advantages of universal adversarial attacks are: they do not need to access the target model during testing; and they significantly reduce the opponent's barrier to entry: the trigger sequence can be widely distributed, and anyone can fool the machine learning model.

3.2. Problem Formulation

Consider a trained text classification model f, a set of benign input text t with truth labels y and correctly predicted by the model $f(t) = y$. Our goal is to connect the found trigger t_{adv} in series with any benign input, which will cause the model f to predict errors, that is, $f(t_{adv}; t) \neq y$.

3.3. Attack Trigger Generation

In order to ensure that the trigger is natural, fluent, and diversified to generate more universal disturbances, we use the Gibbs sampling [19] on a BERT model. This is a flexible framework that can sample sentences from the BERT language model under specific criteria. The input is a customized initial word sequence. In order not to increase the additional restrictions of the trigger, we initialize it to a full mask sequence as in Equation (1).

$$X^0 = (x_1^0, x_2^0, \ldots, x_T^0). \tag{1}$$

In each iteration, we randomly sample a position i uniformly, and then replace the token at the ith position with a mask.The process can be formulated as follows:

$$x_i = [MASK], i = (1, 2, \ldots, T), \tag{2}$$

where $[MASK]$ is a mask token. We get the word sequence at time t, as shown in Equation (3).

$$X_{-i}^t = (x_1^t, \ldots, x_{i-1}^t, [MASK], x_{i+1}^t, \ldots, x_T^t). \tag{3}$$

Then calculate the word distribution p_{t+1} of the language model on the BERT vocabulary according to the Equation (4) and sample a replacement word $\tilde{x_i^t}$ from it.

$$p_{t+1} = \frac{p(x_1^t, \ldots, x_{i-1}^t, y, x_{i+1}^t, \ldots, x_T^t)}{\sum_y p(x_1^t, \ldots x_{i-1}^t, y, x_{i+1}^t, \ldots, x_T^t)}. \tag{4}$$

We used the decision function $h()$ to decide whether to use the proposed word $\tilde{x_i^t}$ or keep the word x_i^{t-1} in the previous iteration. Thus the next word sequence is as in Equation (5).

$$X^t = (x_1^t, \ldots, x_{i-1}^t, \tilde{x_i^t}, x_{i+1}^t, \ldots, x_T^t) \tag{5}$$

We repeated this procedure many times and only select one sample at intervals during the sampling process. After many iterations, we get the desired output. Figure 4 provides an overview framework of our attack algorithm.

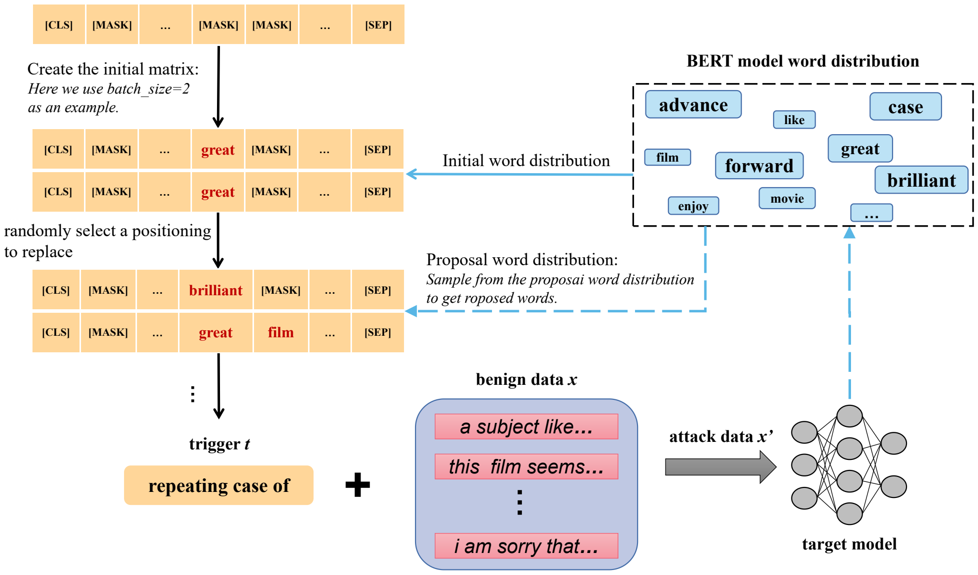
[CLS]
[MASK]
...
[MASK]
[MASK]
...
[SEP]
Create the initial matrix:
Here we use batch_size=2 as an example.
great
randomly select a positioning to replace
brilliant
film
Initial word distribution
BERT model word distribution
advance
like
case
film
forward
great
brilliant
enjoy
movie
...
Proposal word distribution:
Sample from the proposai word distribution to get roposed words.
trigger *t*
repeating case of
benign data *x*
a subject like...
this film seems...
i am sorry that...
attack data *x'*
target model

Figure 4. Overview of our attack. At each step, we concatenate the current trigger to a batch of examples. Then, we sample sentences conditioned on the loss value and classification accuracy computed for the target adversarial label over the batch from a BERT language model.

4. Experiments

In this part, we describe the conducted a comprehensive experiment to evaluate the effect of our trigger generation algorithm on sentiment analysis tasks.

4.1. Datasets and Target Models

We chose two benchmark datasets, including SST-2 and IMDB. SST-2 is a binary sentiment classification data set containing 6920 training samples, 872 verification samples, and 1821 test samples [25]. The average length of each sample is 17 words. IMDB [26] is a large movie review dataset consisting of 25,000 training samples and 25,000 test samples, labeled as positive or negative. The average length of each sample is 234 words. As for the target models, we choose the widely used universal sentence encoding models, namely bidirectional LSTM (BiLSTM).Its hidden states are 128-dimensional, and it uses 300-dimensional pre-trained GloVe [27] word embeddings. Figure 5 provides the BiLSTM framework.

4.2. Baseline Methods

We selected the recent open-source general adversarial attack method as the baseline, and used the same data set and target classifier for comparison [28]. The baseline experiment settings were the same as those in the original paper. Wallace et al. [28] proposed a gradient-guided general disturbance search method. They first initialize the trigger sequence by repeating the word the, subword a, or character a, and connect the trigger to the front/end of all inputs. Then, they iteratively replace the tokens in the triggers to minimize the loss of target predictions for multiple examples.

4.3. Evaluation Metrics

In order to facilitate the evaluation of our attack performance, we randomly selected 500 correctly classified samples in the data set according to the positive and negative categories as the test input. We evaluated the performance of the attack model, including the composite score, the attack success rate, attack effectiveness, and the quality of adversarial examples. The details of our evaluation indicators are listed in Table 1. We will describe these evaluation indicators in detail.

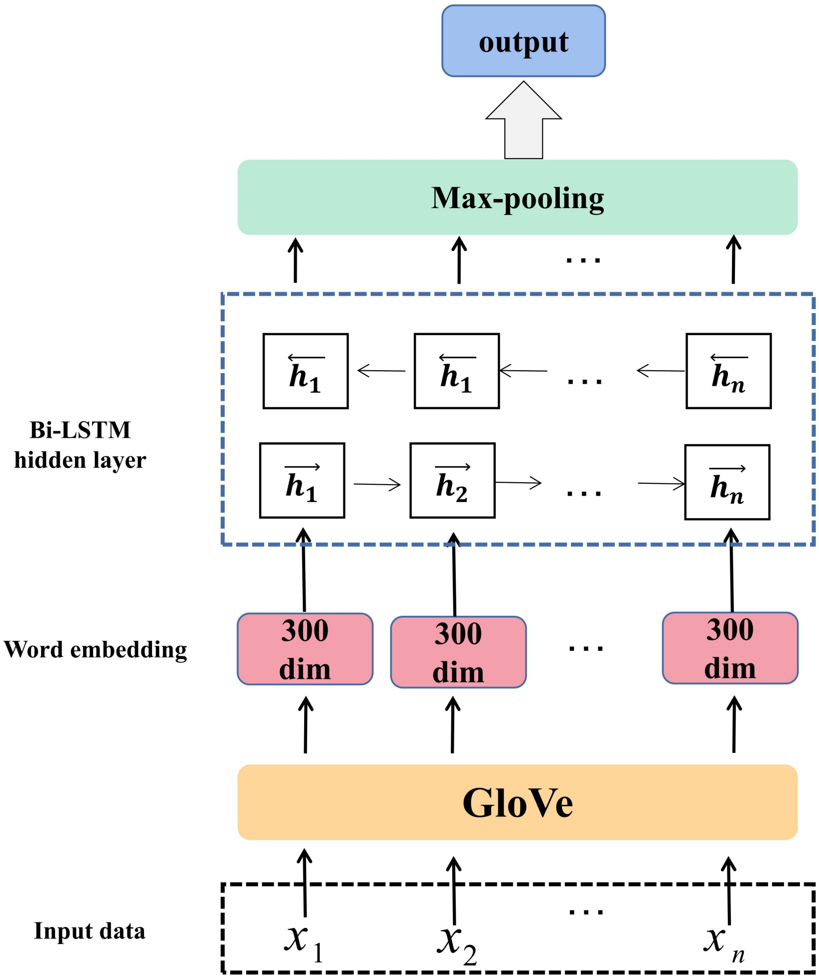

Figure 5. BiLSTM framework.

Table 1. Details of evaluation metrics. "Auto" and "Human" represent automatic and human evaluations respectively. "Higher" and "Lower" mean the higher/lower the metric, the better a model performs.

Metrics	Evaluation Method	Better?
Composite score	Auto	Higher
Success Rate	Auto	Higher
Word Freqency	Auto	Higher
Grammaticality	Auto (Error Rate)	Lower
Fluency	Auto (Perplexity)	Lower
Naturality	Human (Naturality Score)	Higher

(1) The attack success rate is defined as the percentage of samples incorrectly predicted by the target model to the total number of samples. In this experiment, these samples are all connected to the universal trigger. The formula is defined as follows

$$S = \frac{1}{N} \sum_{i=1}^{N} (f(t, x_i) \neq y_i), \tag{6}$$

where N is the total number of samples, f represents the target model, t represents the universal trigger, x_i represents the ith test sample, and y_i represents the actual label of x_i.

(2) We divide it into four parts for the quality of triggers, including word frequency [29], grammaticality, fluency, and naturality [23]. The average frequency of the words in the trigger is calculated using empirical estimates from the training set of the target classifier.

The higher the average frequency of a word, the more times the word appears in the training set. Grammaticality is measured by adding triggers of the same number of words to benign text, and then using an online grammar check tool (Grammarly) to obtain the grammatical error rate of the sentence. With the help of GPT-2 [14], we utilize Language Model Perplexity (PPL) to measure fluency. Naturalness reflects whether an adversarial example is natural and indistinguishable from human-written text.

(3) We construct a composite score Q to comprehensively measure the performance of our attack method. The formula is defined as follows

$$Q = \lambda \cdot S + \mu \cdot W - \nu \cdot G - \delta \cdot P \tag{7}$$

where S is the attack success rate of the trigger, W is the average word frequency of the trigger, G is the grammatical error rate of the trigger, and P is the perplexity of the GPT-2 [14]. W, G, P are all normalized. $\lambda, \mu, \nu, \delta$ is the coefficient of each parameter, and $\lambda + \mu + \nu + \delta = 1$. In order to balance the weight of each parameter, we set λ, μ, ν and δ to 0.25. The higher the Q score, the better the attack performance.

To further verify that our attack is more natural than the baseline, we conducted a human evaluation study. We provide 50 pairs of comparative texts. Each team contains one trigger and one baseline trigger (with or without benign text). Workers are asked to choose a more natural one, and humans are allowed to choose an uncertain option. For each instance, we collected five different human judgments and calculated the average score.

4.4. Attack Results

Table 2 shows the results of our attack and baseline [28]. We observe that our attack achieves the highest composite score Q on all the two datasets, proving the superiority of our model over baselines. For both positive and negative situations, our method has a higher attack success rate. It can be found that the success rate of triggers on SST-2 or IMDB data has reached more than 50%. Furthermore, our method achieved the best attack effect on the Bi-LSTM model trained on the SST-2 data set, with a success rate of 80.1%. Comparing the models trained on the two data sets, the conclusion can be drawn: The Bi-LSTM model trained on the SST-2 data set is the easiest to be attacked by general adversarial attacks.

Table 2. Universal attack results. The composite score Q of our attack is higher than the baseline method. Our attacks are slightly less successful in terms of attack success rate but generate a more natural trigger.

Task	**Test Data**	**Our Attack**			**Baseline**		
		Trigger	**Success Rate (%)**	**Q (%)**	**Trigger**	**Success Rate (%)**	**Q (%)**
SST-2	negative	genius ensemble plays a variety scripts dealing with disease	74.0	**6.25**	death fearlessly courageous courageous terror terror sentimentalizing sentimentalizing triteness	84.3	5.12
	positive	speedy empty constraints both on aimlessly	80.1	**7.78**	wannabe hip timeout timeout ill infomercial	89.1	6.33

Table 2. *Cont.*

Task	Test Data	Our Attack			Baseline		
		Trigger	Success Rate (%)	Q (%)	Trigger	Success Rate (%)	Q (%)
IMDB	negative	harmonica fractured absolutely amazing enjoyable fantasia suite symphony energetically	51.3	**0.15**	unparalleled heartwrenching heartwarming unforgettably wrenchingly movie relatable relatable heartfelt	65.2	−2.20
	positive	red martin on around a keen cherry drinks then limp unfunny sobbing from a waste entrance	50.1	−**0.15**	miserable moron unoriginal unoriginal unengaging ineffectual delicious crappiest stale lousy	57.6	−4.10

Figure 6 shows the comparison of word frequency between benign text and different attack methods. Because a higher word frequency indicates that the word is more common, and a lower frequency indicates that the word is rare. Figure 6 shows that the average word frequency of natural text is the highest. The average word frequency of our trigger is always higher than the baseline method and closer to natural text. Figure 7 compares the Grammarly automatic detection of grammatical error rates when our attack results and baseline results are connected to benign samples simultaneously. Again, it can be seen that our attack has a lower grammatical error rate.

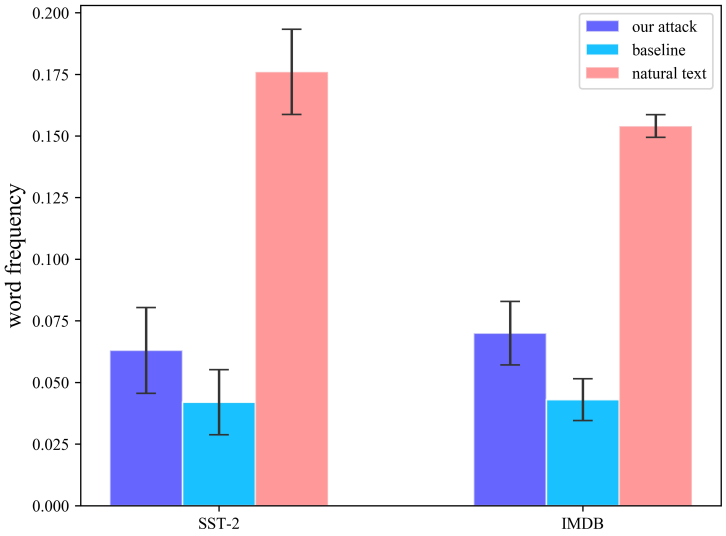

Figure 6. Word frequency. The average frequency and root mean squared error of different triggers in the target model training set (normalized).

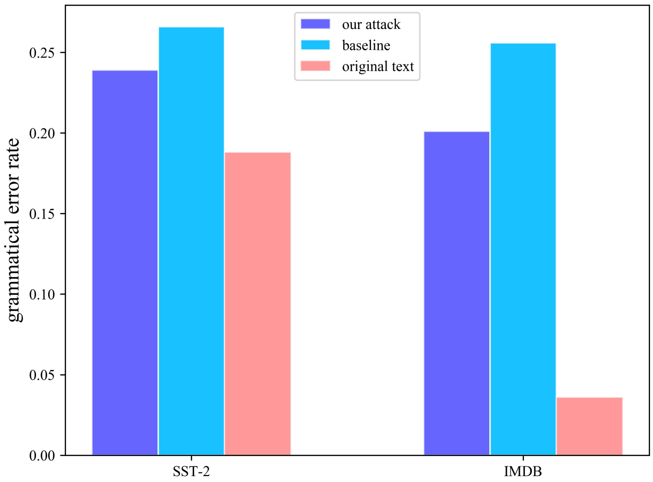

Figure 7. Grammatical error rate in triggers and benign text as the grammar checkers—Grammarly (https://www.grammarly.com) (accessed on 10 October 2021).

In addition, we measure sentence fluency by language model perplexity. Specifically, we evaluated the perplexity of the triggers generated by different methods in the GPT-2 model as shown in Figure 8, and the implementation results show that our trigger has a lower perplexity than the baseline. Therefore, the triggers we generated are better than the baseline method in this comparative information and are closer to the natural text input.

The results of human evaluations are displayed in Table 3. We observed that 78.6% of staff agree that our attack triggers were more natural than the baseline. At the same time, when the trigger is connected to the benign text, 71.4% of people think that our attack is more natural. This shows that our attacks are more natural to humans than the baseline and harder to detect. As we can see from the above discussion, although our trigger is slightly less aggressive than the baseline method, our trigger is more natural, fluent, and readable than the baseline.

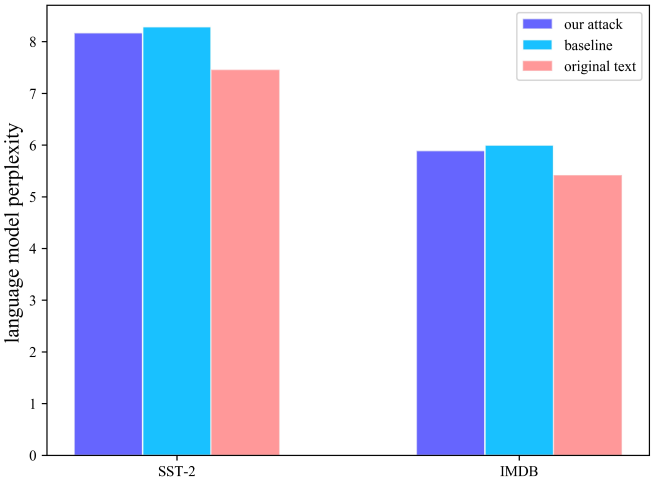

Figure 8. Language model perplexity. We utilize the language model perplexity to measure the fluency with the help of GPT-2 . The y-coordinate is in log-2 scale.

Table 3. Human evaluation results. "Trigger only" means only the text of the trigger sequence. "Trigger + benign" represents sentences where we connect triggers to natural text. "ours" means that our attacks are judged more natural, "baseline" means that the baseline attacks are judged more natural, and "not sure" means that the evaluator is not sure which is more natural.

Condition	Ours	Baseline	Not Sure
Trigger-only	**78.6%**	19.0%	2.4%
Trigger+benign	**71.4%**	23.8%	4.8%

4.5. Transferability

We evaluated the attack transferability of our universal adversarial attacks to different models and datasets. In adversarial attacks, it has become an important evaluation metric [30]. We evaluate the transferability of adversarial examples by using BiLSTM to classify adversarial examples crafted attacking BERT and vice versa. Transferable attacks further reduce the assumptions made: for example, the adversary may not need to access the target model, but instead uses its model to generate attack triggers to attack the target model.

The left side of Table 4 shows the attack transferability of Triggers between different models trained in the sst data set. We can see the transfer attack generated by the BiLSTM model, and the attack success rate of 52.8%∼45.8% has been achieved on the BERT model. The transfer attack generated by the BERT model achieved a success rate of 39.8% to 13.2% on the BiLSTM model.

Table 4. Attack transferability results. We report the attack success rate change of the transfer attack from the source model to the target model, where we generate attack triggers from the source model and test their effectiveness on the target model. Higher attack success rate reflects higher transferability.

Test Class	Model Architecture		Dataset	
	BiLSTM ⇓ BERT	BERT ⇓ BiLSTM	SST ⇓ IMDB	IMDB ⇓ SST
positive	52.8%	39.8%	10.0%	93.9%
negative	45.8%	13.2%	35.5%	98.0%

The right side of Table 4 shows the attack transferability of Triggers between different data sets in the BiLSTM model. We can see that the transfer attack generated by the BiLSTM model trained on the SST-2 data set has achieved a 10.0%∼35.5% attack success rate on the BiLSTM model trained on the IMDB data set. The transfer attack generated by the model trained on the IMDB data set has achieved an attack success rate of 99.9%∼98.0% on the model trained on the SST-2 data set. In general, for the transfer attack generated by the model trained on the IMDB data set, the same model trained on the SST-2 data set can achieve a good attack effect. This is because the average sentence length of the IMDB data set and the amount of training data in this experiment are much larger than the SST2 data set. Therefore, the model trained on the IMDB data set is more robust than that trained on the SST data set. Hence, the trigger obtained from the IMDB data set attack may also successfully deceive the SST data set model.

5. Conclusions

In this paper, we propose a universal adversarial disturbance generation method based on a BERT model sampling. Experiments show that our model can generate both successful and natural attack triggers. Furthermore, our attack proves that adversarial attacks can be more brutal to detect than previously thought. This reminds us that we should pay more attention to the safety of DNNs in practical applications. Future work can explore better ways to best balance the success of attacks and the quality of triggers while also studying how to detect and defend against them.

Author Contributions: conceptualization, Y.Z., K.S. and J.Y.; methodology, Y.Z., K.S. and J.Y.; software, Y.Z. and H.L.; validation, Y.Z., K.S., J.Y. and H.L.; formal analysis, Y.Z. and K.S.; investigation, Y.Z. and K.S.; data curation, Y.Z. and H.L.; writing—original draft preparation, Y.Z., K.S., J.Y. and H.L.; writing—review and editing, Y.Z.; visualization, K.S. and H.L.; supervision, Y.Z., K.S.; project administration, Y.Z. and J.Y.; funding acquisition, J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Anhui Provincial Natural Science Foundation (No.1908085MF202).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The IMDB dataset used to support the findings of the study is public and available at https://s3.amazonaws.com/fast-ai-nlp/imdb.tgz (accessed on 10 October 2021). The SST dataset used to support the findings of the study is public and available at https://nlp.stanford.edu/sentiment/ (accessed on 10 October 2021) .

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1106–1114.
2. Ye, F.; Yang, J. A Deep Neural Network Model for Speaker Identification. *Appl. Sci.* **2021**, *11*, 3603, doi:10.3390/app11083603.
3. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
4. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2014**, arXiv:1312.6199.
5. Papernot, N.; McDaniel, P.; Swami, A.; Harang, R. Crafting Adversarial Input Sequences for Recurrent Neural Networks. *arXiv* **2016**, arXiv:1604.08275.
6. Du, X.; Yu, J.; Yi, Z.; Li, S.; Ma, J.; Tan, Y.; Wu, Q. A Hybrid Adversarial Attack for Different Application Scenarios. *Appl. Sci.* **2020**, *10*, 3559, doi:10.3390/app10103559.
7. Thys, S.; Van Ranst, W.; Goedeme, T. Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Workshops 2019, Long Beach, CA, USA, 16–20 June 2019.
8. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The Limitations of Deep Learning in Adversarial Settings. *arXiv* **2015**, arXiv:1511.07528.
9. Jia, R.; Liang, P. Adversarial Examples for Evaluating Reading Comprehension Systems. *arXiv* **2017**, arXiv:1707.07328.
10. Belinkov, Y.; Bisk, Y. Synthetic and Natural Noise Both Break Neural Machine Translation. *arXiv* **2018**, arXiv:1711.02173.
11. Moosavi-Dezfooli, S.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal Adversarial Perturbations. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 86–94, doi:10.1109/CVPR.2017.17.
12. Wallace, E.; Feng, S.; Kandpal, N.; Gardner, M.; Singh, S. Universal Adversarial Triggers for Attacking and Analyzing NLP. 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 2019, Hong Kong, China, 3–7 November 2019; Inui, K., Jiang, J., Ng, V., Wan, X., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 2153–2162, doi:10.18653/v1/D19-1221.
13. Behjati, M.; Moosavi-Dezfooli, S.; Baghshah, M.S.; Frossard, P. Universal Adversarial Attacks on Text Classifiers. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, UK, 12–17 May 2019; pp. 7345–7349, doi:10.1109/ICASSP.2019.8682430.
14. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. *OpenAI blbog* **2019**, *1*, 9.
15. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.

16. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.G.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R., Eds.; pp. 5754–5764.
17. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
18. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R., Eds.; pp. 3261–3275.
19. Wang, A.; Cho, K. BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model. *arXiv* **2019**, arXiv:1902.04094.
20. Ebrahimi, J.; Rao, A.; Lowd, D.; Dou, D. HotFlip: White-Box Adversarial Examples for Text Classification. *arXiv* **2018**, arXiv:1712.06751.
21. Lei, Q.; Wu, L.; Chen, P.; Dimakis, A.; Dhillon, I.S.; Witbrock, M.J. Discrete Adversarial Attacks and Submodular Optimization with Applications to Text Classification. In Proceedings of the Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, 31 March–April 2 March 2019.
22. Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.J.; Srivastava, M.; Chang, K.W. Generating Natural Language Adversarial Examples. *arXiv* **2018**, arXiv:1804.07998.
23. Zang, Y.; Qi, F.; Yang, C.; Liu, Z.; Zhang, M.; Liu, Q.; Sun, M. Word-level Textual Adversarial Attacking as Combinatorial Optimization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020.
24. Brown, T.B.; Mané, D.; Roy, A.; Abadi, M.; Gilmer, J. Adversarial Patch. *arXiv* **2017**, arXiv:1712.09665.
25. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.; Potts, C. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.
26. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies—Volume 1*; Association for Computational Linguistics: Seattle, WA, USA, 2011; pp. 142–150.
27. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, 25–29 October 2014; A Meeting of SIGDAT, a Special Interest Group of the ACL; Moschitti, A., Pang, B., Daelemans, W., Eds.; ACL: Seattle, WA, USA, 2014; pp. 1532–1543. doi:10.3115/v1/d14-1162.
28. Wallace, E.; Feng, S.; Kandpal, N.; Gardner, M.; Singh, S. Universal Adversarial Triggers for Attacking and Analyzing NLP. *arXiv* **2021**, arXiv:1908.07125.
29. Mozes, M.; Stenetorp, P.; Kleinberg, B.; Griffin, L.D. 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, EACL 2021, Online, 19–23 April 2021; Merlo, P., Tiedemann, J., Tsarfaty, R., Eds.; Association for Computational Linguistics: Seattle, WA, USA, 2021; pp. 171–186.
30. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the physical world. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.