electronics

MDPI

Article

Textual Adversarial Attacking with Limited Queries

Yu Zhang, Junan Yang *, Xiaoshuai Li, Hui Liu and Kun Shao

Institute of Electronic Countermeasure, National University of Defense Technology, Hefei 230037, China; zyu@nudt.edu.cn (Y.Z.); xiaoshuaili@nudt.edu.cn (X.L.); liuhui17c@nudt.edu.cn (H.L.); shaokun20@nudt.edu.cn (K.S.)

* Correspondence: yangjunan@ustc.edu.cn

Abstract: Recent studies have shown that natural language processing (NLP) models are vulnerable to adversarial examples, which are maliciously designed by adding small perturbations to benign inputs that are imperceptible to the human eye, leading to false predictions by the target model. Compared to character- and sentence-level textual adversarial attacks, word-level attack can generate higher-quality adversarial examples, especially in a black-box setting. However, existing attack methods usually require a huge number of queries to successfully deceive the target model, which is costly in a real adversarial scenario. Hence, finding appropriate models is difficult. Therefore, we propose a novel attack method, the main idea of which is to fully utilize the adversarial examples generated by the local model and transfer part of the attack to the local model to complete ahead of time, thereby reducing costs related to attacking the target model. Extensive experiments conducted on three public benchmarks show that our attack method can not only improve the success rate but also reduce the cost, while outperforming the baselines by a significant margin.

Keywords: machine learning; adversarial attack; natural language processing (NLP); black box

Citation: Zhang, Y.; Yang, J.; Li, X.; Liu, H.; Shao, K. Textual Adversarial Attacking with Limited Queries. *Electronics* **2021**, *10*, 2671. https://doi.org/10.3390/electronics10212671

Academic Editor: Amir Mosavi

Received: 2 October 2021
Accepted: 28 October 2021
Published: 31 October 2021

1. Introduction

Despite the impressive performance of deep neural networks (DNNs) in various fields, DNNs are known to be vulnerable to adversarial examples that are maliciously crafted by adding imperceptible interference to humans [1]. This vulnerability has attracted great interest as well as raised great concern about the security of DNNs. As a result, many attack methods have been proposed to further explore the vulnerability of DNNs in image, speech, and text domains [2–6]. However, as text is discrete data, the optimized generation of adversarial examples is difficult, unlike in the case of images. Furthermore, perturbations inserted into text are almost impossible to make genuinely imperceptible. Even the slightest character-level perturbation might drastically alter the semantics of the original input or destroy its fluency.

Textual adversarial attacks can be classified into white- and black-box attacks in terms of the attack setting. In a black-box attack, the attacker can only manipulate the input and output of the target model through query access, without any knowledge about the detailed model structure and parameters [7]. Generally, the efficacy of a black-box attack can be quantified in terms of performance (i.e., attack success rate) and cost, the attack cost is usually measured by the number of model queries required to generate a successful adversarial example.

Existing black-box attacks, from character-level flipping [8] to sentence-level paraphrasing [9], all achieve good performance. In Particular, the word-level attack method based on word replacement performs particularly well in terms of attack efficiency and adversarial example quality [10]. This kind of method can be viewed as a combinatorial optimization problem that combines search space reduction and adversarial example search [11]. Thus, by choosing a suitable search space reduction method and an efficient optimization method, more robust attack performances can be obtained, and higher-quality

adversarial examples can be produced. However, these methods usually require a large number of queries for each generated adversarial example. In addition, the recent rise of large-scale pre-training language model [12] which is well known by bidirectional encoder representations from transformers (BERT) pushed the performance of natural-language-processing (NLP) tasks to new levels. A fine-tuned BERT shows powerful ability in downstream tasks, rendering a confrontational attack on BERT more challenging [13]. Hence, the cost to be paid will be higher.

While the query efficiency of black-box attacks in the text domain has been widely studied, previous studies have not provided targeted research on effective solutions. To solve the problems, we propose a new black-box adversarial attack method whose main idea is to make full use of the adversarial examples generated by the local model, and complete the process of the targeted attack in advance by transferring the part of the process to the local model, which includes three main parts. The first one is an adversarial attack against a local white-box model. A sufficient number of local adversarial examples and corresponding replacement-word positions can be obtained during this step. The second part is the targeted adversarial attack against the target black-box model. In this step, the adversarial examples obtained from the local attack act are regarded as inputs, and the position information of replacement words is used for assistance. The third part uses the tagged by-products obtained from the target attack to tune the local model in real time; this improves the transferability of the local adversarial examples. We conducted detailed experiments to evaluate the proposed method. We also conducted a decomposition analysis to demonstrate the advantages of these three basic ideas.

2. Related Work

Adversarial attacks have been studied extensively in computer vision. Figure 1 briefly shows the basic principle of adversarial attacks. For example, a piece of text labeled 1 can be recognized accurately by DNN model under normal circumstances. However, adding malicious disturbance to this piece of text will lead to DNN model recognition error. Most studies have performed gradient-based perturbations in a continuous input space by using a white-box environment. However, to reduce the criticism and drive toward a more realistic confrontation environment, many researchers have begun to study attack methods under black-box conditions.

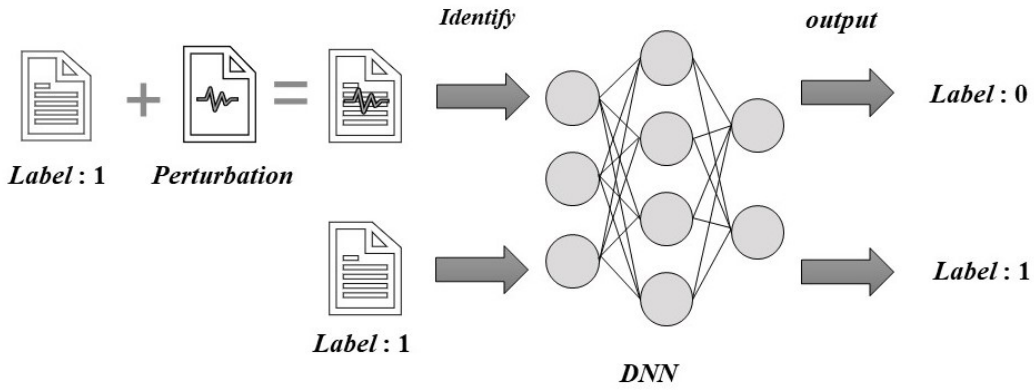

Figure 1. The schematic diagram of adversarial attack.

2.1. Transfer Attacks

Related research has shown that adversarial examples have transferability [14]. That is to say, those generated for the target model may also cause errors in other models trained by using different structures and training sets.

Most studies assumed that the attacker can train a local model by accessing training data similar to those used by the target model. Considering that the training data are difficult to obtain, the attacker marks his/her own data by querying the target model [7,15].

Papernot et al. [16] proposed a reservoir sampling method that improves the training efficiency of local models, thereby reducing costs. Li et al. [15] further reduced the number of queries by adopting active learning. However, even with these advances, there are still too many queries required. In addition, although the adversarial examples can be transferred between models, the success rate of transfer attacks is usually much lower than those of direct attacks.

Most previous studies focused on computer vision, while much less research has been spared for NLP. Gil et al proposed a transfer-based adversarial attack method for the text domain [17]. Despite its high efficiency and low cost, it is challenging to achieve satisfactory attack effects using the proposed method, which achieved a 42% attack success rate against the Google Perspective application program interface. Our experiment conducted on an IMDB dataset obtained a success rate of only 11.1% for migrating adversarial examples from the local model to the target model, whereas the success rate of direct attacks was almost 100%.

2.2. Direct Attacks

Instead of attack transference, direct attacks provide a way of directly attacking the target model. There are many methods for generating adversarial examples for deep-learning model attacks, such as the fast-gradient sign attack [18], Jacobian-based saliency map attack [3], and Deepfool method [19]. However, these methods are aimed at image fields and cannot be directly applied to text. This is because an image comprises continuous data, whereas text is discrete and has word-order restrictions. Furthermore, the distance metric used to measure the distance between text and images is different. According to the perturbation degree of adversarial examples, existing text adversarial attack models can be divided into three categories, described as follows.

Sentence-level attacks include the addition of distracting sentences [20], paraphrased content, and perturbations [21] in a continuous latent semantic space. In addition, by using one-hot character embedding [8] or optical character embedding [22], gradient-based character-replacement methods have been explored. The adversarial examples made by these methods are usually different from the original input samples. Thus, their effectiveness cannot be guaranteed.

Character-level attacks mainly involve random-character operations, including exchanges, substitutions, deletions, insertions, and repetitions [23,24]. While the success rate of character-level attacks is high, it destroys the grammar and the structure of the original input. However, such attacks are easily defended [25].

For word-level attacks, most methods consist of two parts: Search-space reduction for locating adversarial examples and the adversarial example-space reduction method, which involves word-embedding [26] or language models [27] to filter words. This may include selecting synonyms as substitutes [13,28,29], combinations [30,31], or sememe-based word-replacements [11]. Search algorithms include gradient descent [26,32], genetic algorithms [30], Metropolis–Hastings sampling [27], saliency-based greedy algorithm [29,33], and discrete particle-swarm [11]. While the aforementioned methods have achieved good results in terms of attack effectiveness and sample quality, the costs were not considered.

3. Methodology

In this section, we propose a new general word-level advanced attack framework which make use of the information of the adversarial examples generated by the local model and to transfer part of the process to the local model for completion ahead of time.

3.1. Threat Model

Under black-box conditions, the attacker cannot access the model directly and has no knowledge of the internal parameters of the model. It can query the target model only by the input samples provided to obtain the prediction result. In the study of transfer attacks, it is assumed that the adversary has access to a pre-trained local model. There are

two main ways to obtain these local models, one is by using model distillation technique, the other is by using training data similar to those used in the target model. In model distillation technology, a large number of queries must be made to the target model, and the prediction results of the target model are used as labels to obtain a batch of training data for training the local model. In the latter way, the local model is trained directly after the data meeting the attack conditions is obtained, and the target model does not need to be repeatedly queried. However, the attack success rate is low in this way due to the strong attack hypothesis.

3.2. Basic Ideas

This framework is based on three basic ideas, listed as follows.

The use of transferability of adversarial examples: Research shows that neural-network models based on different structures may misclassify the same adversarial example, i.e., the adversarial example is transferable. This is because different models used for similar tasks often have similar decision boundaries [14]. Therefore, we first conducted an adversarial attack against a local model to obtain candidate adversarial examples which is expected to be transferred to the target model. While the candidate adversarial examples generated on the local model cannot be completely transferred, they are still more aggressive than unprepared examples. Therefore, we considered candidate adversarial examples that fail to transfer as the next starting point for attacks against the target.

The use of location information of vulnerable words searched by local models: Because different models may have similar decision boundaries for similar tasks, the same sample may suffer the same word vulnerabilities based on textual expression characteristics, regardless of whether the model is local or target; this can cause decision errors in the model. Therefore, we considered the positions of vulnerable words found when each local model generates candidate adversarial examples, especially those that fail to transfer successfully. This information is passed to the next step of the attack process, during which the costs of researching vulnerable words are reduced.

The use decision information of the target model to optimize the local model in real time: Continue to attack the target model on the basis of the candidate adversarial samples to obtain the final disturbance samples. The prediction information of these samples may contain richer information about the decision boundary of the target model. Therefore, we used these disturbed samples, including successful adversarial examples and failed texts, to tune the local model in real time to achieve a model closer to the target model.

3.3. Attack Method

Our advanced attacks combine the three aforementioned ideas according to word-level confrontational attacks. We made full use of the prior information obtained by the local model while building adversarial examples, including candidate examples along with their keyword position information. We used the labeled input received during the target-model attack to optimize the local model to continuously improve the transfer rate. The proposed advanced attack method is shown in Algorithm 1 and is composed of three main steps. In addition, in Figure 2 we show the overall flow of our method.

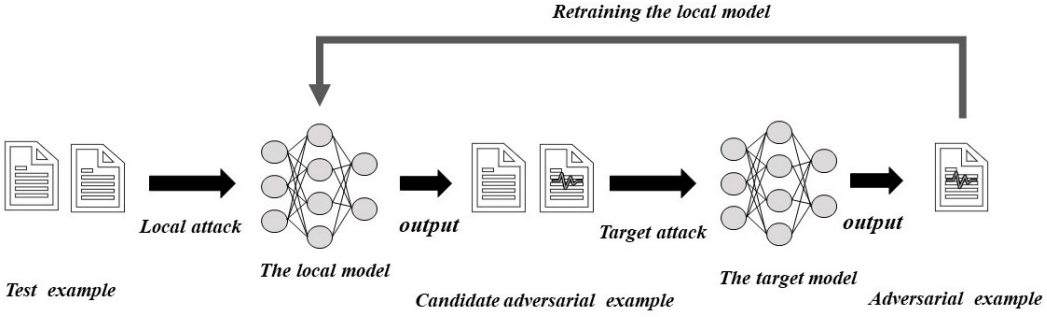

Figure 2. Overview of our method.

Algorithm 1 Advance Attack

Input: Sample text $X = (x_1, x_2, \cdots, x_n)$, the corresponding ground-truth label, Y, the local model, $L()$, and the target black-box model, $F()$
Output: Adversarial example X_{adv}

Initialization $X^{'} \leftarrow X$, $T^{'} \leftarrow X$
while each sentence x_i in X **do**
Determine the candidate replacement word space according to the sememe of each word in x_i.
$\omega_m^i \leftarrow (\omega_m^1, \omega_m^2, \cdots, \omega_m^n)$
Find the candidate word having the highest prediction score $L()$ for each word in x_i.
$W_{max} = (\omega_m^r, \omega_j^p, \cdots, \omega_k^q)$
Optimize the combination to find a suitable replacement
$x_i^{'} = (\omega_1, \omega_2, \cdots, \omega_j^p, \cdots, \omega_k^q, \cdots, \omega_n)$
$p_i = (j, \cdots, k)$(Replacement word position)
if $F(x_i^{'}) = y$ **then**
Calculate $\omega_m^{'i}$ using the highest prediction score of $F()$ at position p_i in $x_i^{'}$
$W_{max}^{'} = (\omega_j^e, \cdots, \omega_k^f)$
$x_i^{''} = (\omega_1, \omega_2, \cdots, \omega_j^e, \cdots, \omega_k^f, \cdots, \omega_n)$
if $F(x_i^{''}) = y$ **then**
In $x_i^{''}$ words other than p_i position continue to be combined and optimized
end if
else
return X_{adv}, X_t.
end if
$T.insert(X_t)$
end while
return X_{adv}

Local attacks: The attack process generates adversarial examples against the local model. Specifically, the given input text, $x_i = (\omega_1, \omega_2, \cdots, \omega_m, \cdots, \omega_n)$, contains n words , each with a variable number of replaceable words in the selected search space (e.g., thesaurus, semantics, or word-embedding space). For example, the set of replacement words for the word, ω_m, can be expressed as $(\omega_m^1, \omega_m^2, \cdots, \omega_m^n)$. We then queried the local model to find the candidate replacement word with the highest target-label prediction score in each replaceable word space, $W_{max} = (\omega_m^r, \omega_j^p, \cdots, \omega_k^q)$, where ω_m^r is the optimal replacement word for the m^{th} word in sentence x_i. For the local model, we used combinatorial optimization methods to screen out the appropriate optimal replacement word combination. In addition, we used the combination to replace the word at the corresponding position of the original sentence, thereby generating a candidate adversarial example, $x_i^{'}$. This process can be repeated to obtain the required number of candidate adversarial examples. Then, we recorded the position information, $p_i = (j, \cdots, k)$, of the replacement words of each candidate example.

Target Attack: The attack process generates adversarial examples against the target models. If the candidate adversarial example directly transfers, it continues to attack the next example. If the example fails to transfer directly, the candidate adversarial example is used as a starting point to attack the target model. According to Step 1, the candidate adversarial examples that fail to transfer are still closer to the target model's decision boundary than the original examples. Therefore, the candidate adversarial example, $x_i^{'}$, which failed to transfer, uses the replacement word position information, $p_i = (j, \cdots, k)$, to find the word's replaceable word space from the j^{th} to the k^{th} position in $x_i^{'}$. By querying the target model, we obtained the candidate replacement word, $W_{max}^{'}$, by using the target

model's highest prediction score and directly replace it from the j^{th} to the k^{th} position in $x_i^{'}$ with $(\omega_j^e, \cdots, \omega_k^f)$ to obtain $x_i^{''}$. If the attack is successful, the iteration ends; otherwise, we continue to select replacement words for the remaining positions.

Tuning the local model: After performing word replacement on the original input example, if the target model can be deceived, a successful adversarial e xample, x_{adv}, is returned. The adversarial examples obtained during the target attack, regardless of success, obtain the prediction score of the target model during the search process. We added these examples, X_t, and the target model prediction label to the training dataset of the local model to further train it.

4. Experiments

In this section, we conduct the experiments conducted to evaluate the attack framework. Section 4.1 introduces the dataset and model settings, Section 4.2 introduces the attack settings, and Section 4.4 provides the experimental results. We primarily focused on the degree of reduction in attack costs which are measured by the number of queries required to generate each adversarial example.

4.1. Datasets and Models

For sentiment analysis, we selected two benchmark datasets: IMDB [34] and SST [35]. Both are binary sentiment classification datasets. The IMDB dataset is a large film review dataset containing 25,000 training samples and 25,000 test samples. The average length of per sample is 234 words. The SST-2 is a Stanford sentiment tree library with 6920 training samples, 872 validation samples, and 1821 test samples. The average length per sample is 17 words. Both the IMDB and SST-2 data sets are classified into two categories, namely positive and negative. The mean sentence length of the SST-2 dataset is much smaller than that of IMDB dataset. This makes adversarial attacks on the SST-2 dataset more difficult. We also experimented with the natural language inference task, where we used Stanford Natural Language Inference (SNLI) dataset. The task was to judge the relationship between two sentences and decide whether the second sentence can be derived from an entailment, contradiction, or neutral relationship with the first [36].

For the local model, we chose a general sentence-coding model with a simple structure and good performance by using a bidirectional long short-term memory with max pooling [37]. The target black-box model uses the powerful pretraining language model, BERT, which is a pretrained masking-language model for large-scale unsupervised general language knowledge. Details of the datasets and classification accuracy results of the original models are listed in Table 1.

Table 1. Details of datasets and accuracy observed using original models. "Train," "Val," and "Test" refer to the instance numbers of the training, validation, and test sets, respectively. "BiLSTM %ACC" and "BERT %ACC" are the original classification accuracies of BiLSTM and BERT, respectively.

Dataset	Task	Avg Len	Train	Val	Test	BiLSTM %ACC	BERT %ACC
IMDB	Sentiment Analysis	234	25,000	0	25,000	89.19	90.76
SST-2	Sentiment Analysis	17	6920	872	1821	83.52	90.30
SNLI	NLI	8	550,152	10,000	10,000	84.92	89.51

4.2. Attack Configuration

The word-level adversarial attack under black-box conditions can be regarded as a combinatorial optimization problem. This study used this scenario as a premise for the related study, which includes two main steps. The first step entails reducing the search space. The existing methods mainly include a replacement word search using a synonym

dictionary, or they utilize a replacement word search to find adjacent words in the word vector space. Another common method is the sememe-based word substitution method. The second step involves a search for adversarial examples, and this is executed using particle-swarm optimization, greedy, and genetic algorithms.

With extant word-level attack methods, in terms of success and quality, the combination of the sememe-based word substitution method and particle-swarm optimization (SEM-PSO) is considered the best. However, the cost of attacking is higher.

Therefore, to test the superiority of the proposed method and evaluate whether the method can effectively reduce attack costs without sacrificing performance, SEM-PSO was compared with previous works.

4.3. Evaluation Metrics

Based on a previous study [30], we randomly selected 1000 samples correctly classified by both local and target models from the three cited datasets as input. We used the attack cost for evaluation while attending to the attack success rate. The attack cost was measured by the number of queries required for each adversarial example.

4.4. Attack Performance

To evaluate the performance of our proposed method more comprehensively, we compared the experimental results of our method with the latest relevant research results, as shown in Table 2. We measure the attack cost by the average number of queries required for each adversarial sample generated by the attack target model, specifically, in the target attack phase, the total number of queries spent on all samples divided by the number of adversarial samples successfully found. As observed, our attack method achieves the highest attack success rate while requiring the least number of queries to the IMDB and SST-2 datasets and the lowest modification rate. Owing to the particularity of the SNLI dataset, we cannot perform tuning experiments on this model. We only performed the first two steps of our proposed method. As a result, our attack success rate for SNLI has not increased significantly, and is lower than that of TextFooler. This demonstrates that the proposed method improves upon previous work.

Table 2. Attack results of different attack models and automatic evaluation results of the adversarial example quality. Queries/AE represents the average number of queries for each successful adversarial example of a target attack.

Dataset	Attack Method	Attack Success Rate(%)	Queries/AE	Modif Rate(%)
IMDB	Genetic	55.0	778.5	3.94
	PWWS	81.0	1287.3	20.65
	TextFooler	81.0	1134.0	6.10
	Ours	99.3	418.2	6.07
SST-2	Genetic	54.0	226.2	18.38
	PWWS	87.0	122.9	23.70
	TextFooler	77.9	57.9	27.15
	Ours	92.3	81.3	10.96
SNLI	Genetic	83.5	613.0	20.80
	PWWS	-	-	-
	TextFooler	94.1	60.0	18.50
	Ours *	84.0	42.6	18.31

* The performance of SNLI is the result of not tuning the local model.

As shown in Table 3, advanced attacks dramatically reduce costs while slightly improving the success rate. The existing method could reduce the average number of queries for all examples and average number of queries for each successful example by more than 50% on the IMDB, respectively. Our method can reduce the average number of queries for all examples by 53%, and the average number of queries for each successful example was reduced by 55%. Additionally, the attack success rate increased from 91.1% to 92.3%

for SST-2. Similarly, our attack strategy showed superior performance on SNLI. There are two main reasons for the cost reduction: Some candidate adversarial examples transferred directly, whereas others did not. However, they were useful for staging new attacks. This is analyzed in more detail in Section 4.5.

Table 3. The adversarial attacks directly against the target model starting with the original sample are the original stacks, while the advanced attacks starting with local adversarial examples are ours. Queries/All represents the average query number of all input samples in the target attack. Queries/AE represents the average number of queries for each successful adversarial example of a target attack. Because the Stanford Natural Language Inference (SNLI) dataset requires cumbersome processing, and the cost of each attack and retraining is high, we cannot perform tuning experiments on that model. We could test only the first two ideas.

Dataset	Attack Method	Attack Success Rate(%)	Queries/All	Queries/AE
IMDB	Original	99.5	878.6	838.4
	Ours	99.3	437.9	418.2
SST-2	Original	91.1	331.5	181.2
	Ours	92.3	154.0	81.3
SNLI	Original	79.5	312.6	101.3
	Ours *	84.0	142.1	42.6

* The performance of SNLI is the result of not tuning the local model.

4.5. Decomposition Analyses

We conducted a detailed analysis of the three basic steps of the proposed method, aiming to further verify the advantages of our advanced attack.

4.5.1. Selection of Input Examples for Target Attack

Table 4 shows the attack cost and corresponding attack success rates using the original and local adversarial examples as starting points. "Transfer + original" represents how the candidate adversarial examples generated for the local model were transferred; those that failed to transfer still used their original examples as starting points for attack. "Transfer + Candidate" represents a method of attacking by using those local adversarial examples that failed to transfer as starting points.

The experimental results show that the success rate of our proposed attack method based on "Transfer + candidate" on the SST-2 dataset increased by 1.0%, and the number of queries for each successful example was reduced by 34.1% compared with those of direct attacks. Simultaneously, the attack performance and attack costs were superior to those to the "Transfer + original" attack method. The proposed attack method based on "Transfer + candidate" has a slightly higher attack success rate on the IMDB data set. Simultaneously, the average number of queries for each sample in the test set and the average number of queries for successfully attacked examples was reduced by 22.7% and 20.5%, respectively, when compared to the "Transfer + original" method. Furthermore, on the SNLI dataset the attack success rate for "Transfer + candidate" significantly increases relative to "Transfer + original", while the number of queries required gradually decreases. This is sufficient to show that although the local adversarial examples could not be fully transferred to the target model, these local adversarial examples were still more aggressive than the original ones.

4.5.2. Impact of Vulnerable Words

To detect the benefits of the vulnerable-word position to the attacker, we recorded the position information of the replacement word of each candidate adversarial example generated during the local attack relative to the original example. Then, during the target attack process, the perturbation was preferentially selected at the recorded position of the candidate adversarial example. If the target model could not be fooled after disturbing

these positions, we continued implementing the disturbance to the remaining positions until a termination condition was reached. "Transfer + Position" in Table 4 represents the addition of vulnerable word position information relative to "Transfer + Candidate".

The addition of vulnerable-word information improved the performance of our method. The average query cost of successful examples was reduced by 17.3%, 17.7%, and 8.7% on the three datasets, respectively, indicating that the vulnerable words causing model judgment errors in some samples were probably the same, whether in the local or target models.

Table 4. Impact of starting from local adversarial examples and keyword location information.

Dataset	Attack Method	Transfer Rate(%)	Attack Success Rate(%)	Queries/All	Queries/AE
IMDB	Transfer + Original	11.1	99.5	781.1	744.9
	Transfer + Candidate	11.1	99.7	603.1	592.0
	Transfer + Position	11.2	99.7	510.8	489.0
SST-2	Transfer + Original	25.0	91.1	256.2	136.0
	Transfer + Candidate	25.0	91.9	261.3	120.1
	Transfer + Position	25.0	92.0	235.1	98.8
SNLI	Transfer + Original	32.2	79.5	211.9	60.3
	Transfer + Candidate	32.2	83.8	146.7	46.7
	Transfer + Position	32.2	84.0	142.1	42.6

4.5.3. Local Model Tuning

To verify that the tags learned from the target attack could be used to tune the local model, we measured the change in the transfer rate of the local model after tuning as well as the impact on the number of queries. Algorithm 1 shows that the local model will be updated every time an adversarial example is obtained in target attack. However, we update the model every once in a while to consider the computational cost of tuning. For the SST and IMDB datasets, we updated the model for every 200 samples. Because SNLI requires cumbersome data processing, and the cost of each attack and retraining is high, we could not perform tuning experiments on it.

In order to detect the transferability of the tuned local model, 1000 samples were randomly selected from the test set, and candidate adversarial examples were obtained by adversarial attacks against the local model. Then these samples were input into the black box target model to test the mobility of candidate adversarial examples. The candidate adversarial examples were used as the starting point for new attacks and to observe query costs. We first verify the changes in mobility of the local model before and after fine-tuning to test whether the local model can be updated using tagged samples obtained from target attacks. The first experiment helped us examine the utility of fine-tuning the local model without worrying about other possible interactions. The second experiment showed that improving the fine-tuning of the local model could benefit the attacker.

Figure 3 shows the results of the first experiment. After the model was tuned, the transfer rate of the two datasets gradually increased. In the SST dataset, when the number of feedback samples reached 600, the transfer rate stabilized at approximately 26%, and the number of queries to the target model was reduced. The transfer rate increased from the initial 10.6% to 13.4% on the IMDB dataset. These results verify that the use of disturbance samples to tune the local model is valid.

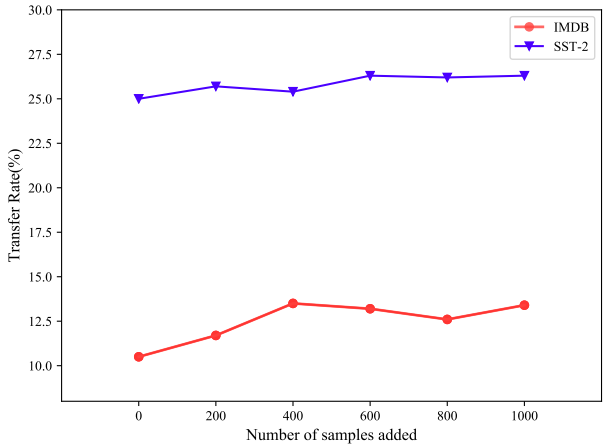

Figure 3. Transfer-rate changes with sample update.

Table 5 lists the results of the second experiment, showing the combination of model tuning with the first and second steps. The experiments showed that the transferability improvement obtained through model tuning dramatically reduced the cost of attack. For example, for the "Transfer + Position" attack of the SST dataset, the average query cost of successful samples was reduced by 17.7%, i.e., from 98.8 to 81.3, and the attack success rate increased from 92.0% of the static local model to 92.3%. For the "Transfer + Candidate" attack on the SST dataset and the "Transfer + Position" attack on the IMDB dataset, although the attack success rate was reduced by 0.5% and 0.4%, the average query cost of successful samples was reduced by 5% and 14%, which are within the tolerable range.

Table 5. Impact of tuning local models.

Dataset	Attack Method	Success Rate (%)		Queries/All		Queries/AE	
		Static	Tuned	Static	Tuned	Static	Tuned
IMDB	Transfer + Candidate	99.7	99.7	603.1	541.9	592.0	528.2
	Transfer + Position	99.7	99.3	510.8	437.9	489.0	418.2
SST-2	Transfer + Candidate	91.9	91.4	261.3	218.9	120.1	113.6
	Transfer + Position	92.0	92.3	235.1	154.0	98.8	81.28

5. Conclusions and Future Work

In this paper, we proposed a new black-box text adversarial attack strategy that used the information of adversarial examples generated by a local model. We conducted extensive experiments to demonstrate the superiority of our model in terms of attack cost, attack success rate, and adversarial example quality. Our proposed advanced attack strategy offers a significant improvement over the most advanced results in query cost, thus providing a more accurate estimate of the attacker's cost in a black box setting.

Nevertheless, when an adversarial attack is performed on a local model, the obtained adversarial examples are transferred to the target model with a low original transfer rate; this affects the further improvement of the attack. Thus, the analysis of the similarity between the models to improve the transferability of the original model could be one possible solution to perfect the method in the future.

Author Contributions: Conceptualization, Y.Z., J.Y. and X.L.; methodology, Y.Z., J.Y. and K.S.; software, Y.Z. and H.L.; validation, Y.Z., X.L., J.Y., K.S. and H.L.; formal analysis, Y.Z., X.L. and K.S.; investigation, Y.Z. and X.L.; 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 research received no external funding.

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 (28 April 2020). The SST dataset used to support the findings of the study is public and available at https://nlp.stanford.edu/sentiment/accessed on (28 April 2020). The SNLI dataset used to support the findings of the study is public and available at https://nlp.stanford.edu/projects/snli/ accessed on (29 April 2020).

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

References

1. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.J.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014; Bengio, Y., LeCun, Y., Eds.; Conference Track Proceedings.
2. Thys, S.; Ranst, W.V.; Goedemé, T. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; pp. 49–55. [CrossRef]
3. Papernot, N.; McDaniel, P.D.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The limitations of deep learning in adversarial settings. In Proceedings of the IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, 21–24 March 2016; pp. 372–387. [CrossRef]
4. Li, J.; Ji, S.; Du, T.; Li, B.; Wang, T. TextBugger: Generating adversarial text against real-world applications. In Proceedings of the 2019 Network and Distributed System Security Symposium, San Diego, California, USA, 24–27 February 2019. [CrossRef]
5. Yu, F.; Wang, L.; Fang, X.; Zhang, Y. The defense of adversarial example with conditional generative adversarial networks. *Secur. Commun. Netw.* **2020**, *2020*, 3932584:1–3932584:12. [CrossRef]
6. Jiang, L.; Qiao, K.; Qin, R.; Wang, L.; Yu, W.; Chen, J.; Bu, H.; Yan, B. Cycle-Consistent Adversarial GAN: The integration of adversarial attack and defense. *Secur. Commun. Netw.* **2020**, *2020*, 3608173:1–3608173:9. [CrossRef]
7. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; Swami, A. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, 2–6 April 2017.
8. Ebrahimi, J.; Rao, A.; Lowd, D.; Dou, D. HotFlip: White-box adversarial examples for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July 2018; Gurevych, I., Miyao, Y., Eds.; Volume 2: Short Papers, pp. 31–36. [CrossRef]
9. Iyyer, M.; Wieting, J.; Gimpel, K.; Zettlemoyer, L. Adversarial example generation with syntactically controlled paraphrase networks. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, LA, USA, 1–6 June 2018; Walker, M.A., Ji, H., Stent, A., Eds.; Volume 1 (Long, Papers), pp. 1875–1885. [CrossRef]
10. Wang, W.; Wang, R.; Wang, L.; Wang, Z.; Ye, A. Towards a robust deep neural network against adversarial texts: A survey. *IEEE Trans. Knowl. Data Eng.* **2021**, 1. [CrossRef]
11. 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.
12. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Volume 1 (Long and Short, Papers), pp. 4171–4186. [CrossRef]
13. Jin, D.; Jin, Z.; Zhou, J.T.; Szolovits, P. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, 7–12 February 2020; pp. 8018–8025.
14. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into transferable adversarial examples and black-box attacks. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
15. Li, P.; Yi, J.; Zhang, L. Query-Efficient Black-Box Attack by Active Learning. In Proceedings of the IEEE International Conference on Data Mining, ICDM 2018, Singapore, 17–20 November 2018; pp. 1200–1205. [CrossRef]

16. Papernot, N.; McDaniel, P.D.; Goodfellow, I.J. Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. *arXiv* **2016**, arXiv:1605.07277.
17. Gil, Y.; Chai, Y.; Gorodissky, O.; Berant, J. White-to-Black: Efficient distillation of black-box adversarial attacks. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Volume 1 (Long and Short, Papers), pp. 1373–1379. [CrossRef]
18. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**, arXiv:1412.6572.
19. Moosavi-Dezfooli, S.; Fawzi, A.; Frossard, P. DeepFool: A simple and accurate method to fool deep neural networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582. [CrossRef]
20. Jia, R.; Liang, P. Adversarial examples for evaluating reading comprehension systems. *arXiv* **2017**, arXiv:1707.07328.
21. Zhao, Z.; Dua, D.; Singh, S. Generating natural adversarial examples. *arXiv* **2018**, arXiv:1710.11342.
22. Eger, S.; Sahin, G.G.; Rücklé, A.; Lee, J.; Schulz, C.; Mesgar, M.; Swarnkar, K.; Simpson, E.; Gurevych, I. Text processing like humans do: Visually attacking and shielding NLP systems. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1 (Long and Short Papers), pp. 1634–1647.
23. Belinkov, Y.; Bisk, Y. Synthetic and natural noise both break neural machine translation. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018.
24. Gao, J.; Lanchantin, J.; Soffa, M.L.; Qi, Y. Black-Box generation of adversarial text sequences to evade deep learning classifiers. In Proceedings of the 2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, 24 May 2018; pp. 50–56. [CrossRef]
25. Pruthi, D.; Dhingra, B.; Lipton, Z.C. Combating adversarial misspellings with robust word recognition. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; Korhonen, A., Traum, D.R., Màrquez, L., Eds.; Volume 1: Long, Papers, pp. 5582–5591. [CrossRef]
26. Sato, M.; Suzuki, J.; Shindo, H.; Matsumoto, Y. Interpretable adversarial perturbation in input embedding space for text. *arXiv* **2018**, arXiv:1805.02917.
27. Zhang, H.; Zhou, H.; Miao, N.; Li, L. Generating fluent adversarial examples for natural languages. *arXiv* **2020**, arXiv:2007.06174.
28. Samanta, S.; Mehta, S. Towards crafting text adversarial samples. *arXiv* **2017**, arXiv:1710.11342.
29. Ren, S.; Deng, Y.; He, K.; Che, W. Generating natural language adversarial examples through probability weighted word saliency. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1085–1097.
30. Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.J.; Srivastava, M.B.; Chang, K.W. Generating natural language adversarial examples. *arXiv* **2018**, arXiv:1804.07998.
31. Glockner, M.; Shwartz, V.; Goldberg, Y. Breaking NLI systems with sentences that require simple lexical inferences. *arXiv* **2018**, arXiv:1805.02266.
32. Papernot, N.; Mcdaniel, P.; Swami, A.; Harang, R.E. Crafting adversarial input sequences for recurrent neural networks. In Proceedings of the MILCOM 2016-2016 IEEE Military Communications Conference, Baltimore, MD, USA, 1–3 November 2016; pp. 49–54.
33. Liang, B.; Li, H.; Su, M.; Bian, P.; Li, X.; Shi, W. Deep text classification can be fooled. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018. [CrossRef]
34. 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, Stroudsburg, PA, USA, 19–24 June 2011; Association for Computational Linguistics, HLT '11: Seattle, WA, USA; pp. 142–150.
35. 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; Association for Computational Linguistics: Seattle, WA, USA, 2013; pp. 1631–1642.
36. Bowman, S.R.; Angeli, G.; Potts, C.; Manning, C.D. A large annotated corpus for learning natural language inference. *arXiv* **2015**, arXiv:1508.05326.
37. Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; Bordes, A. Supervised learning of universal sentence representations from natural language inference data. *arXiv* **2017**, arXiv:1705.02364.