applied sciences

MDPI

Article

WordBlitz: An Efficient Hard-Label Textual Adversarial Attack Method Jointly Leveraging Adversarial Transferability and Word Importance

Xiangge Li, Hong Luo * and Yan Sun

School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China; lixiangge@bupt.edu.cn (X.L.)
* Correspondence: luoh@bupt.edu.cn

Abstract: Existing textual attacks mostly perturb keywords in sentences to generate adversarial examples by relying on the prediction confidence of victim models. In practice, attackers can only access the prediction label, meaning that the victim model can easily defend against such hard-label attacks by denying access based on the attack's frequency. In this paper, we propose an efficient hard-label attack approach, called WordBlitz. First, based on the adversarial transferability, we train a substitute model to initialize the attack parameter set, including a candidate pool and two weight tables of keywords and candidate words. Then, adversarial examples are generated and optimized under the guidance of the two weight tables. During optimization, we design a hybrid local search algorithm with word importance to find the globally optimal solution while updating the two weight tables according to the attack results. Finally, the non-adversarial text generated during perturbation optimization is added to the training of the substitute model as data augmentation to improve the adversarial transferability. Experimental results show that WordBlitz surpasses the baseline in terms of better effectiveness, higher efficiency, and lower cost. Its efficiency is especially pronounced in scenarios with broader search spaces, and its attack success rate on a Chinese dataset is higher than on baselines.

Keywords: natural language processing; textual attack; hard label; adversarial samples; model robustness

check for updates

Citation: Li, X.; Luo, H.; Sun, Y. WordBlitz: An Efficient Hard-Label Textual Adversarial Attack Method Jointly Leveraging Adversarial Transferability and Word Importance. *Appl. Sci.* **2024**, *14*, 3831. https://doi.org/10.3390/app14093831

Academic Editor: João M. F. Rodrigues

Received: 18 March 2024
Revised: 26 April 2024
Accepted: 28 April 2024
Published: 30 April 2024

1. Introduction

Deep Neural Networks (DNNs) have achieved unprecedented successes in the fields of text, image, and speech processing. However, their security concerns have attracted widespread attention. Studies have indicated that DNNs are vulnerable to threats from adversarial examples [1]. Attackers can mislead the classifier with subtle perturbations which are imperceptible to humans. Thus, the study of how to generate adversarial examples is essential in exploring the robustness of DNNs.

In NLP tasks, attackers primarily generate adversarial examples by perturbing keywords in the input text, which is known as word-level attack. These attacks are broadly categorized as white-box attacks and black-box attacks based on the information available to the attacker [2]. In black-box attacks, the attacker can only obtain the output of the target model, which makes such attacks more challenging. Previous research has mainly focused on score-based attacks [3], which estimate the importance of words by relying on the class probabilities or confidence score of the target model. For instance, the importance of each word in a sentence can be evaluated through the change in model output results after word deletion. However, it is difficult to obtain the score of target models in real-world applications. Thus, a few studies have only leveraged the label for attacks, which are termed decision-based or hard-label attacks [4].

In recent years, research in deep learning has been evolving towards Large-scale Language models (LLMs). LLMs are complex and have a high number of parameters; meaning that attackers sometimes cannot access the output scores of LLMs. Consequently, we believe that hard-label attacks will become more significant in the future. Existing research in this area can be categorized based on whether it addresses traditional attacks or transfer-based attacks.

- Traditional attacks utilize genetic algorithms, word importance, etc., to generate adversarial examples. These methods share a common drawback in that each round requires random initialization and cannot use the historical attack experience of other texts. Hence, these attack methods heavily rely on the impact of random initialization [5], making them are inefficient, as they require frequent access to the target model. The victim model can simply detect this based on its frequency.
- Transfer-based attacks assume that adversarial examples designed for one victim model are likely to fool other victim models as well [6]. This characteristic is known as adversarial transferability. In this approach, a substitute model is trained using the input and output of the victim model. In this way, a hard-label attack is transformed into a decision-based attack or white-box attack. The advantage lies of this approach lies in its high efficiency and less need for access to the victim model. However, the success rate is sometimes low.

Transfer-based attacks are particularly suitable in the hard-label scenario, where access to the victim model is restricted. Numerous recent studies have focused on applying adversarial transferability to attack DNNs [7]. This research has indicated several challenges. (1) **Structural Disparities in Models**: Transfer-based attacks do not work well when there is a significant structural difference between the victim model and substitute model. (2) **Weaker Classification Performance of Substitute Model**: In general, stronger classifiers lead to better adversarial transferability; however, in the hard-label scenario, the attacker cannot train the substitute model with enough data due to the limitations on access to the victim model, meaning that its performance is not as good. (3) **Poor Generalization Performance of Substitute Model**: A number of studies have suggested that the data augmentation can enhance generalization performance and improve adversarial transferability [8]. How to enhance generalization performance with data augmentation remains a challenge. (4) **Adversarial Overfitting Due to Muti-step Attacks**: Research has indicated that multi-step attacks are more likely than single-step attacks to overfit to the parameters of victim model. Due to the discrete nature of textual data, attack algorithms based on keyword replacement typically require multiple iterations, which reduces the effectiveness of transfer-based attacks.

To address the above issues, we propose a hard-label attack method named WordBlitz that jointly leverages word importance and adversarial transferability. First, we train a substitute model with limited training data from the victim model and extract the word importance. Then, the Attack Parameter Set (APS), including the candidate pool and word importance, is constructed from the substitute model based on adversarial transferability. Subsequently, adversarial examples are generated and optimized from the APS by a hybrid local search algorithm. Meanwhile, the APS is updated based on the attack result. Furthermore, we utilize the text generated in previous attacks in data augmentation to enhance the generalization of the substitute model. WordBlitz effectively addresses the issues of poor transferability caused by model structural difference, poor classification and overfitting due to multi-step attacks, and the need for improved universality of attack algorithms with respect to victim models. Our contributions are as follows:

- We propose a transfer-based attack method for the hard-label scenario and change the target of transferability from label to word importance. This approach provides a heuristic method to initialize the attack parameter set from the substitute model.
- We introduce an efficient perturbation strategy relying on the attack parameter set. The attack parameter set enables quick searching for optimal results, and is updated with the attack results to leverage historical experience.

- We propose a data augmentation method for the substitute model by utilizing the text generated in the perturbation, which improves the adversarial transferability from history. Hence, our approach performs more significantly in broad search spaces, such as Chinese datasets.

2. Related Work

Score-based Attacks. Earlier studies employed the score-based attack method, in which attackers can obtain the class probabilities or confidence score of victim model. Most of these first find important words which highly impact the confidence score of the victim model; these keywords are then replaced with candidate pool. For example, Li et al. [9] selected and masked the vulnerable words with a masked language model. Then, BERT (Bidirectional Encoder Representations from Transformers) was used to predict the masked words and generate adversarial examples. Similarly, Garg and Ramakrishnan [10] presented BAE, which replaces and inserts tokens in the original text by masking a portion of the text and leveraging BERT-MLM to generate alternatives for the masked tokens. Li et al. [11] proposed a contextualized adversarial samples generation model and produced fluent outputs through a mask-then-infill procedure. Maheshwary et al. [12] introduced a query-efficient attack strategy which jointly leverages an attention mechanism and locality-sensitive hashing to reduce the query count. Lee et al. [13] focused on using Bayesian optimization to improve attack efficiency. Unlike the strategies, other works have used optimization procedures to craft adversarial inputs. For example, Alzantot et al. [14] proposed a population-based optimization algorithm to fool well-trained sentiment analysis and textual entailment models.

Decision-based Attacks. Only a few existing attack methods are decision-based. Zhao et al. [15] first proposed a perturbation method in the continuous latent semantic space. Ribeiro [16] rewrote sentences following semantically equivalent adversarial rules. Maheshwary et al. [5,12] leveraged a population-based optimization algorithm called HLBB to craft plausible and semantically similar adversarial examples, only relying on the top label predicted by the target model. Ye et al. [4] proposed a gradient-based hard-label method named TextHoaxer that estimates the gradient directly from the virtual randomly sampled directions in the embedding space rather than from concrete candidate words. Yu et al. [17] proposed TextHacker, which adopts a hybrid local search algorithm and estimates the word importance from history to minimize perturbation. Due to the strict query budget constraints in hard-label scenarios, decision-based attack methods need to show improved attack efficiency. However, these methods typically suffer from low attack efficiency, requiring frequent access to the output results of the victim model. For example, the HLLB method that uses genetic algorithms cannot utilize historical attack experience, while TextHacker employs a random initialization strategy. Hence, when the query budget is low, the attack success rates significantly decrease. Improving attack efficiency is a key research issue that needs to be addressed.

Transfer-based Attacks. Transfer-based attacks set the victim model as the target model and assume that adversarial examples can transfer between different models. These methods rely on the training data from the target model. Vijayaraghavan and Roy [18] trained a substitute model to mimic the decision boundary of the target classifier, then generated adversarial examples against the substitute model and transferred them to target model. However, the adversarial transferability is sometimes poor, resulting in a low attack success rate. Therefore, a few studies have explored the factors affecting this issue [7]. Most related works explain transferability from a model perspective, claiming that the decision boundary [19], model architecture [20], or generalization performance of the substitute model [21] significantly influence the adversarial transferability. Another consideration is that the attack algorithm may impact the adversarial transferability. Xie et al. [8] found that multi-step attacks tends to generate more overfitted adversarial perturbations with lower transferability than single-step attacking. Wang et al. [22] explained this based on the interactions inside adversarial perturbations. Motivated by these studies, a few researchers have aimed to improve the transferability.

For example, Wang et al. [23] introduced data augmentation to improve the generalization performance of the substitute model and enhance the transferability.

3. Methodology

3.1. Formula Descriptions

To make the definitions more comprehensible, we provide some formula explanations about textual classifiers and adversarial samples.

Textual classifier. A textual classifier can be defined as a function $F : X \rightarrow Y$ which maps an input textual set X to a label set Y, where Y is a set of labels, for example, {positive, negative} or a more extensive set of labels.

Adversarial samples. Attackers aim to adding a small perturbation e in the original text x to generate an adversarial example x'. Here, x' misleads the classifier F such that $F(x) \neq F(x')$; meanwhile, the generated x' should be imperceptible. A series of metrics can be adopted to achieve this goal, such as $\|\varepsilon\| < \delta$, where δ is a threshold used to limit the size of perturbations.

3.2. Framework

The structure of WordBlitz is shown as the main framework in Figure 1, which revolves around the Attack Parameter Set (APS). WordBlitz consists of two modules, namely, Adversarial Initialization and Perturbation Optimization, along with a data augmentation process. The APS is constructed from the pretrained substitute model and updated based on the attack result on the target (victim) model. It aims to save the importance weights of each word in the input text and candidate pools. The adversarial examples are generated and optimized relying on the APS. In detail, the input text is first sent to the adversary initialization module. We replace vulnerable words with the weight table in APS to quickly generate the initial adversarial example. Next, the initial adversarial examples enter the perturbation optimization module, where hybrid local search operations are used to find the optimal result. Finally, the generated non-adversarial samples are used for data augmentation of the substitute model to improve its generalization and classification performance while enhancing the adversarial transferability.

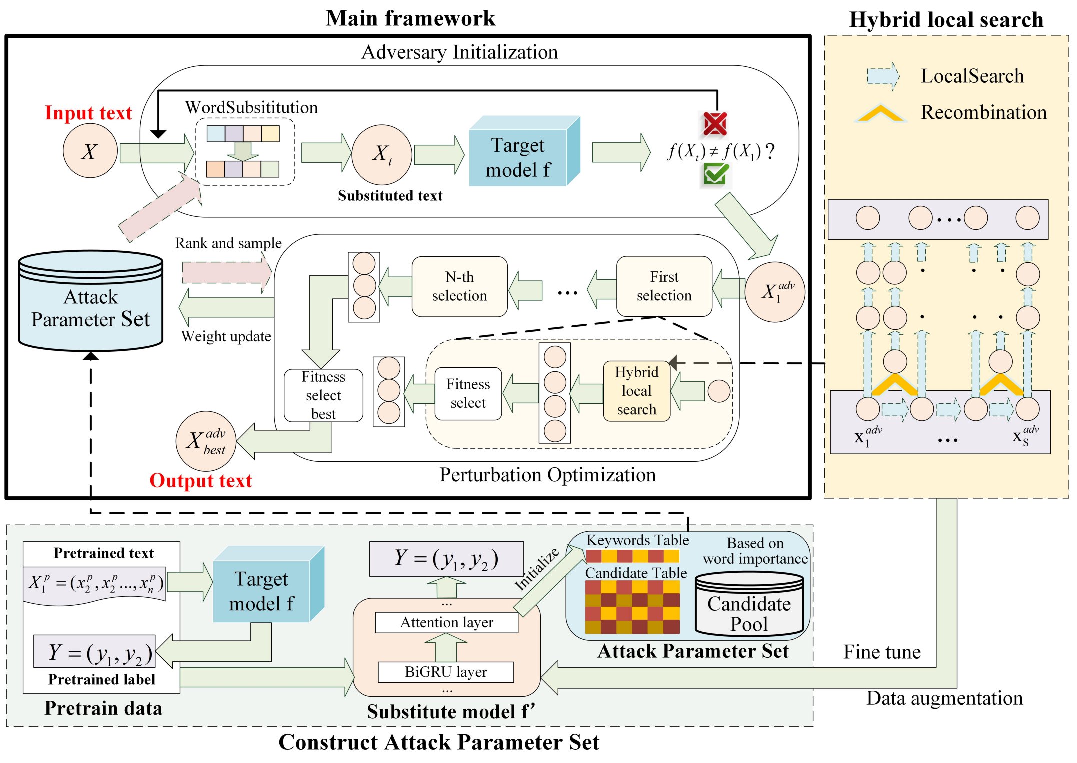


Figure 1. The framework of WordBlitz.

3.3. Construction of the APS

The APS is the fundamental component of WordBlitz, including a candidate pool and Attack Weight Table (AWT). The candidate pool provides similar words for replacement to generate adversarial examples. Within the AWT, the position weight table documents vulnerable keyword weights, while the candidate table documents replacement word weights from candidate pool. To enhance the perturbation efficiency, we initialize the position weight table by training a substitute model. The entire construction process comprises three stages: constructing the candidate pool, training the substitute model, and initializing the AWT.

3.3.1. Candidate Pool

The candidate pool is constructed based on language features and common attack methods. In the case of English, we utilize near-synonyms to form a candidate pool, which is then employed to generate perturbation through near-syn. For Chinese, we construct the candidate pool using shape-close characters, near-phonetic words, and split-character dictionaries, which are respectively utilized to generate perturbation with pinyin, glyphs, and character splitting. For a given word x_i, its candidate pool is donated by $C(x_i) = (\hat{x}_i^0, \hat{x}_i^1, \cdots, \hat{x}_i^m)$.

3.3.2. Substitute Model

We set the victim model as the target model f. Due to the unstable efficiency of random or constant initialization, it is essential to devise a heuristic method to initialize the position weight table reasonably and achieve efficient perturbation on keywords. Similarly, the word weights of the attention mechanism should accurately capture each word's extent of influence. Hence, we train a BiGRU model with an attention mechanism as the substitute model f' in order to initialize the position weight table in the AWT. These words can then be perturbed using the word importance. The training data are obtained from the input and output of the target model f. The input text $X = (x_1, \cdots x_i, \cdots, x_n)$ is first encoded by the BiGRU to obtain the hidden state $H = (h_1, \cdots h_i, \cdots h_n)$, with the i-th hidden state h_i computed as follows:

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

Then, the attention matrix $\alpha = (\alpha_1, \cdots, \alpha_i, \cdots, \alpha_n)$ is calculated with H, with α_i formulated a

$$u_i = \tanh(W_{\text{word}}\, h_i + b_{\text{word}}\,), \tag{2}$$

$$\alpha_i = \frac{\exp(u_i^\top u_i)}{\sum_{j=1}^n \exp\left(u_j^\top u_j\right)}, \tag{3}$$

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

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

After the fully connected layer and softmax layer, the probability distribution p for classification is obtained. We use the negative log-likelihood of the correct labels as the training loss L, formulated as

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

$$L = -\sum_k \log p_{kj}, \tag{6}$$

where W_{fc} represents the weight of fully connected layer, b_{fc} is the bias, k is the amount of input text, and j is the label of X. The whole structure of f' is shown in Figure 2.

The substitute model f' is pretrained with only a small amount of training data obtained from the target model f. After each round of the attack, the generated non-adversarial text is utilized for data augmentation to update the parameters of f' with the aim of improving the adversarial transferability.

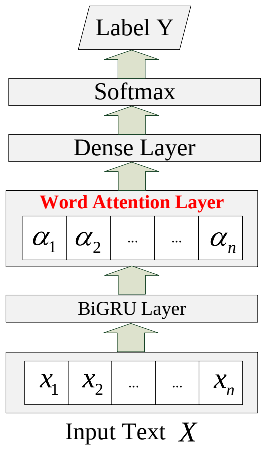


Figure 2. Structure of the substitute model f'.

3.3.3. AWT Matrix

To perturb the keywords efficiently, we attempt to identify the replaced priority of words in both the sentence and candidate pool. These two priorities are stored in a position weight table and a candidate weight table, respectively.

- **Position Weight Table** $W^p = \left(w_1^p, \cdots, w_i^p, \cdots, w_n^p\right)$. Each word and word position in the sentence influences the classification result. Thus, perturbing words with higher importance has a significant impact on the overall semantics. Considering that the word weight in an attention mechanism reflects the contribution of each word in a sentence, we initialize the position weight table W^p approximately with $\alpha = (\alpha_1, \cdots, \alpha_i, \cdots, \alpha_n)$ from the substitute model f'; then, W^p is updated based on the result of hybrid local search in order to better adapt to the target model f.
- **Candidate Weight Table** $W^c = \left[W_1^c, \ldots, W_i^c, \ldots, W_n^c\right]$. We construct W^c to store the substitution scores of each word in the candidate pool. The matrix has the shape $(n, m+1)$, where n represents the number of words in the sentence and $m+1$ denotes the number of words in each candidate pool. W^c is initialized with all ones and updated based on the hybrid local search results.

3.4. Adversary Initialization

WordBlitz generates the initial adversarial examples in adversary initialization based on the APS. We design a ***WordSubstitution*** operator to substitute keywords in a sentence. To narrow down the search space, we replace words based on their w_i^p ranking within the top 50% with random words in the candidate pool. For X_t at the t-th iteration, we perturb it based on W^p and the candidate pool C to craft a new text X_{t+1}, which can be denoted as follows:

$$X_{t+1} = \text{WordSubsitution}(X_t, C, W^p). \tag{7}$$

Iteration is repeated until X_{t+1} is an adversary or reaches the maximum number T.

3.5. Perturbation Optimization

We apply the hybrid local search algorithm [24] for perturbation optimization. Hybrid local search is a kind of population-based algorithm which is effective for combinational optimization problems. It usually contains two key components, namely, ***LocalSearch*** and ***Recombination***. The ***LocalSearch*** operator searches for a better one from the neighborhood of each solution to approach the local optima, while the ***Recombination*** operator crosses over the existing solutions to accept non-improved solutions, helping it jump out of local optima. Inspired by TextHacker [17], we utilize an additional components called WeightUpdate in WordBlitz to allow it to learn from history.

3.5.1. LocalSearch

For an adversarial example X_t^{adv} at the t-th iteration, we randomly sample several (at most k) less important words $\hat{x}_i^{j_t} \in X_t^{\mathrm{adv}}$ from W^{p} with probability p_i:

$$p_i = \frac{1-\sigma\left(w_i^{\mathrm{p}}\right)}{\sum_{i=1}^{n}\left[1-\sigma\left(w_i^{\mathrm{p}}\right)\right]} \tag{8}$$

where $\sigma(\mathrm{x}) = 1/(1+e^{-x})$ is the sigmoid function. This reduces excessive gaps and makes the probability more reasonable. Then, each chosen word $\hat{x}_i^{j_t}$ is equally substituted with the original word $\hat{x}_i^0$ or a candidate word $\hat{x}_i^{j_{t+1}}$ in $C(x_i)$ based on probability $p_{i,j_{t+1}}$ to generate a new text X_{t+1}^{adv}. Here, $p_{i,j_{t+1}}$ is calculated as follows:

$$p_{i,j_{t+1}} = \frac{\sigma\left(w_{i,j_{t+1}}^{\mathrm{c}}\right)}{\sum_{j_{t+1}}^{m}\sigma\left(w_{i,j_{t+1}}^{\mathrm{c}}\right)}. \tag{9}$$

We accept X_{t+1}^{adv} if it is still adversarial; otherwise, we add it to set D for data augmentation and return X_t^{adv}. LocalSearch utilizes W^{p} and W^{c} to substitute less important words with the original word or a candidate word, with the aim of optimizing the perturbation from the k-neighborhood of X_t^{adv}.

3.5.2. WeightUpdate

In order to better highlight the important words in the sentence and candidate pool, we update the word importance W^{p} and W^{c} with X_t^{adv} and X_{t+1}^{adv} using the following rules:
Rule1: For each replaced word $\hat{x}_i^{j+1}$ in $C(x_i)$, if X_{t+1}^{adv} is still adversarial, it means that $\hat{x}_i^{j+1}$ has a positive influence on the adversary. Its weight is increased with reward r_{c} as

$$w'^{c}_{i,j_{t+1}} = w^{\mathrm{c}}_{i,j_{t+1}} + r_{\mathrm{c}} \tag{10}$$

or reduced as

$$w'^{c}_{i,j_{t+1}} = w^{\mathrm{c}}_{i,j_{t+1}} - r_{\mathrm{c}}. \tag{11}$$

Rule2: For each operated word and its position i, if X_t^{adv} is still adversarial, it means that the position i has less influence on the adversary. Thus, its weight is decreased with negative reward $-r_{\mathrm{p}}$ as

$$w'^{\mathrm{p}}_i = \frac{n-r_{\mathrm{p}}}{n}w_i^{\mathrm{p}} \tag{12}$$

or increased as

$$w'^{\mathrm{p}}_i = \frac{n+r_{\mathrm{p}}}{n}w_i^{\mathrm{p}}, \tag{13}$$

where n is the number of words in sentence X.

3.5.3. Recombination

LocalSearch can find a better adversarial example; however, it is likely to result in local optima. ***Recombination*** can effectively address this issue. We randomly combine two randomly sampled texts $X^a = \left(x_1^a, x_2^a, \cdots, x_n^a\right) \in \mathcal{P}^t$ and $X^b = \left(x_1^b, x_2^b, \cdots, x_n^b\right) \in \mathcal{P}^t$ to generate a recombined text $X^c = \left(x_1^c, x_2^c, \cdots, x_n^c\right)$. Each word x_i^c in X^c is randomly sampled from $\left\{x_i^a, x_i^b\right\}$ based on their weights in W^{c}. We repeat this operation $O/2$ times and return all samples, where O is the number of input adversarial examples. Then, we select fitness based on the modified number and semantic similarity with the USE [25]. In summary, ***Recombination*** accepts non-improved solutions in order to avoid local optima.

3.6. DataAugmentation

Due to limited access to target model, we train the substitute model f' with sparse data. This results in poor classification and generalization performance of f', in turn leading to poor adversarial transferability. Considering that there are a large amount of texts generated over multiple iterations, we design a ***DataAugmentation*** operator that adds those generated texts which are non-adversarial to the training data of f as data augmentation to make it stronger. This adjustment enables APS to remember the features of the target model, thereby enhancing the adversarial transferability.

3.7. The Whole Algorithm

The overall algorithm of WordBlitz is summarized in Algorithm 1. It comprises four fundamental steps: Construct APS (lines 4–8 in Algorithm 1, detailed in Section 3.3), Adversary Initialization (lines 10–16 in Algorithm 1, detailed in Section 3.4), Perturbation Optimization (lines 18–31 in Algorithm 1, detailed in Section 3.5), and Data Augmentation (lines 33–37 in Algorithm 1, detailed in Section 3.6).

Algorithm 1 The WordBlitz Algorithm

```
1: **Input:** Input text $X$, target model $f$, query budget $T$, Population Size S, maximum number of local search $N$,
   attention weight $\alpha$, similarity threshold $\varepsilon$, perturb rate threshold $\epsilon$, the set of samples for DataAugmentation
   $D_{\text{da}}$
2: **Output:** Attack result and adversarial example
3: ▷ **Construct APS (line 4–8)**
4: **for** each word $x_i$ in $X$ **do**
5:     Construct the candidate pool $C(x_i)$
6: $Y_{\text{pre}} = f(X_{\text{pre}})$
7: Train substitute model $f'$ with $\{X_{\text{pre}}, Y_{\text{pre}}\}$
8: Initialize $W^{\text{p}}$ with Attention Weight $\alpha$ in $f'$; Initialize $W^{\text{c}}$ with all 1's        ▷ Initialize APS
9: ▷ **Adversary Initialization (line 10–16)**
10: $X_1 = X, X_1^{\text{adv}} = \text{None}$
11: **for** $t = 1$ to $T$ **do**
12:     $X_{t+1} = \textbf{\textit{WordSubstitution}}(X_1, C, W^{\text{p}})$        ▷ Detailed in Section 3.4
13:     **if** $f(X_{t+1}) \neq f(X)$ **then**
14:         $X_1^{\text{adv}} = X_{t+1}$; break        ▷ Initialization succeeds
15: **if** $X_{\text{adv}}$ is None **then**
16:     **return** False, None        ▷ Initialization fails
17: ▷ **Perturbation Optimization (line 18–31)**
18: $\mathcal{P}^1 = \{X_1^{adv}\}$
19: **for** $i = 1$ to $S - 1$ **do**
20:     $X_{i+1}^{\text{adv}} = \textbf{\textit{LocalSearch}}(X_i^{\text{adv}}, C(x_i), W^{\text{c}}, W^{\text{p}})$; $\mathcal{P}^1 = \mathcal{P}^1 \cup \{X_{i+1}^{adv}\}$
21: $t = t + S - 1; g = 1$
22: **while** $t \leq T$ **do**
23:     $\mathcal{P}^g = \mathcal{P}^g \cup \{ \textbf{\textit{Recombination}}\ (\mathcal{P}^g, W_c)\}$        ▷ Detailed in Section 3.5.3
24:     **for** each text $X_{adv_g}$ in $P_g$ **do**
25:         $X_{adv} = X_{adv_g}$
26:         **for** $i = 1$ to $N$ **do**
27:             $X_{i+1}^{\text{adv}} = \textbf{\textit{LocalSearch}}(X_i^{\text{adv}}, C(x_i), W^{\text{p}}, W^{\text{c}})$        ▷ Detailed in Section 3.5.1
28:             $\textbf{\textit{WeightUpdate}}(X_i^{adv}, X_{i+1}^{\text{adv}}, W^{\text{p}}, W^{\text{c}}, f)$        ▷ Update APS based on attack result, detailed in Section 3.5.2
29:         $\mathcal{P}^g = \mathcal{P}^g \cup \{X_{N+1}^{adv}\}; t = t + N$
30:     Construct $\mathcal{P}^{g+1}$ with the top $S$ fitness based on the Modified Number and USE
31:     Record the global optimal $X_{\text{best}}^{\text{adv}}$ based on the Modified Number and USE
32: ▷ **Data Augmentation (line 33–37)**
33: Construct $D_{\text{text}}$ with all generated Text $X_{\text{ge}}$ in Perturbation Optimization
34: **for** each $X_{\text{ge}}$ in $D_{\text{text}}$ **do**
35:     **if** $X_{\text{ge}}$ is non-adversarial **then**
36:         $D_{\text{da}} = D_{\text{da}} \cup \{X_{\text{ge}}, f(X_{\text{ge}})\}$        ▷ Construct set for data augmentation
37: $\textbf{\textit{DataAugmentation}}(f', D_{\text{da}})$
38: **if** $\textbf{\textit{Similarity}}(X_{\text{best}}^{\text{adv}}, X) < \varepsilon$ and $\textbf{\textit{PerturbRate}}\ (X_{\text{best}}^{\text{adv}}, X) < \epsilon$ **then**
39:     return success,$X_{\text{best}}^{\text{adv}}$        ▷ Attack succeeds
40: **return** False,None        ▷ Attack fails
```

4. Experiments

We conducted extensive experiments on three English datasets to validate the effectiveness of WordBlitz. In order to validate its applicability on different languages, we also conducted experiments on three Chinese datasets and additionally incorporated a language model optimized for Chinese.

4.1. Experimental Setup

Victim Models. We adopted TextCNN [26], LSTM [27], and BERT [28] as the English victim models, while adopting TextCNN, BERT, and ERNIE [29] as the Chinese victim models. In addition, to verify the attack effect of WordBlitz on LLMs, we added ChatGLM3-6B [30] as a common victim model.

Baselines. We used two hard-label attack methods, HLBB [5] and TextHacker [17], as our baselines. HLBB uses a population-based algorithm to generate adversarial examples, while TextHacker perturbs the text based on word importance and hybrid local search.

Evaluation Metrics. The evaluation metrics included two aspects, namely, attack effectiveness and perturbation cost. Attack effectiveness encompasses the attack success rate, attack stability, and attack efficiency, while perturbation cost includes the semantic similarity and perturbation ratio. We provide detailed descriptions of these metrics in the corresponding sections.

Datasets. We adopted three English datasets (IMDB [31], Yelp [32], and MR [33]) and three Chinese datasets (Waimai, OnlineShopping, and hotels from ChnSentiCorp) for text classification. These datasets are commonly used for sentiment classification tasks; each data point consists of a user review, and each review is labeled as either “positive” or “negative”. As perturbing keywords in the reviews can effectively change their sentiment polarity, these datasets are widely used in experiments involving adversarial examples. Table 1 provides detailed information about the datasets.

Table 1. The details of datasets.

Dataset	Language	Total Number of Samples	Description
IMDB	English	50,000	Movie Reviews
Yelp	English	94,000	Reviews from Yelp
MR	English	10,662	Movie Reviews
Hotel	Chinese	10,000	Hotel Reviews
Waimai	Chinese	18,000	Reviews for Take-out Food
Online_Shopping	Chinese	60,000	Reviews for Online Shopping

4.2. Evaluation of Attack Effectiveness

We first conducted evaluations using the above datasets on the models under the same query budget of 2000. The evaluation metrics were the attack success rate (Succ, %), perturbation rate (Pert, %), and semantic similarity (Sim, %) as evaluated with Universal Sentence Encoder [25]. Tables 2 and 3 show the results on the English and Chinese datasets, respectively.

As shown in Table 2, WordBlitz performs better than HLBB and TextHacker. HLBB exhibits the weakest attack efficiency due to its reliance on a population optimization algorithm, which cannot learn from history; meanwhile, its inability to calculate candidate word importance leads to a higher perturbation rate. TextHacker introduces word importance and updates it from history to generate and optimize the adversarial examples. Therefore, its attack efficiency is higher than that of HLBB. WordBlitz constructs the APS and initializes it based on the adversarial transferability rather than random initialization, enabling it to generate adversarial examples more quickly. Thus, it achieves a higher attack success rate and semantic similarity with lower perturbation across almost all of the datasets and victim models. Taking the results of attacks on BERT as an example, WordBlitz outperforms the others in terms of attack success rate by clear margins of 0.3–7.9%, and improves the perturbation rate by 0.1–1.7% and semantic similarity by 2.6–3%.

Table 2. Evaluation on attack effectiveness on English datasets.

Model	Attack	IMDB			Yelp			MR		
		Succ.	Pert.	Sim.	Succ.	Pert.	Sim.	Succ.	Pert.	Sim.
TextCNN	HLBB	74.0	4.2	85.3	67.1	7.6	86.2	71.1	13.2	**84.3**
	TextHacker	77.8	**3.0**	85.6	75.4	6.4	82.5	78.3	**11.1**	82.1
	WordBlitz	**78.3**	3.1	**87.1**	**75.9**	**6.3**	**86.4**	**78.9**	11.8	84.0
LSTM	HLBB	72.1	4.1	86.4	61	6.6	85.3	68.3	11.2	83.5
	TextHacker	76.2	3	84.9	65.4	5.5	84.8	75.2	11.2	82.9
	WordBlitz	**76.9**	**3**	**86.8**	**66**	**5.3**	**86.6**	**75.5**	**10.7**	**84.8**
BERT	HLBB	77	4.8	84.1	57.1	8.2	84.6	65.8	11.6	85.2
	TextHacker	81.5	3.4	83.1	63.2	6.7	82.2	73.1	11.4	83.6
	WordBlitz	**81.9**	**3.3**	**85.7**	**63.5**	**6.5**	**85.2**	**73.7**	**10.9**	**85.4**
ChatGLM3	HLBB	36.4	7.1	83.6	35.2	8.4	86	35.6	12.2	81.7
	TextHacker	37.3	6.2	84.2	35.9	7.3	83.4	36.1	**11.3**	82.3
	WordBlitz	**40.4**	**5.9**	**85.1**	**39.8**	**7**	**86.1**	**37.9**	11.5	**83.6**

The bold number denotes the best performance value for each dataset.

Table 3. Evaluation of attack effectiveness on Chinese datasets.

Model	Attack	OnlineShopping			Waimai			Hotel		
		Succ.	Pert.	Sim.	Succ.	Pert.	Sim.	Succ.	Pert.	Sim.
TextCNN	HLBB	43.5	20.7	79.2	46.1	19.5	81.3	47.2	20.4	**81.0**
	TextHacker	52.3	**19.1**	80.4	54.2	**18.1**	79.9	55.8	**19.4**	79.1
	WordBlitz	**59.8**	**18.5**	**81.5**	**60.4**	**18.3**	**83.2**	**62.1**	**18.5**	**82.4**
BERT	HLBB	39.4	18	81.9	43.9	19.2	81.7	41.5	19.1	81.6
	TextHacker	50.2	17.9	81.1	52.6	19.9	81.3	53.2	18.3	80.3
	WordBlitz	**60.1**	**17.4**	**83**	**60.3**	**18.9**	**83.8**	**59.6**	**17.6**	**81.9**
ERNIE	HLBB	41.6	19.2	80.8	42.5	20.3	82.6	44	18.9	81.7
	TextHacker	50.4	18.6	80.2	51.8	19.6	81	54.1	18.6	81.2
	WordBlitz	**58.9**	**18.3**	**82.5**	**59.4**	**19.4**	**82.9**	**60.4**	**17.8**	**83.3**
ChatGLM3	HLBB	32.5	18.9	79.3	33.7	18.4	81.2	31.8	17.6	81.4
	TextHacker	32.7	18.3	81.6	35.1	19.7	79.4	34.3	**18**	80.2
	WordBlitz	**34.3**	**17**	**82.4**	**35.7**	**18.1**	**81.5**	**36.8**	**17.4**	**83**

The bold number denotes the best performance value for each dataset.

Unlike the English scenario, there are more attack types in the Chinese candidate pool, which leads to a broader search space. This situation challenges the attack efficiency. As shown in Table 3, the gap between the three methods is more significant on the Chinese datasets. WordBlitz and TextHacker, which can perturb vulnerable words based on word importance, are more efficient than HLBB. In particular, WordBlitz initializes the APS based on transferability rather than random initialization, allowing it achieve a 1.6–9.1% higher attack success rate than TextHacker and a 1.8–20.7% higher success rate than HLBB. These results show that WordBlitz performs better on complex languages.

4.3. *Stability Evaluation*

To consider the influence of random seeds on the three tested methods, we validated their stability by conducting repeated experiments on the Yelp and Hotel datasets with the

BERT model. For the stability evaluation, we used the coefficient of variation γ, which can be formulated as

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(z_i - \bar{z})^2}, \tag{14}$$

$$\gamma = \frac{S}{\bar{z}}, \tag{15}$$

where s is the standard deviation, z_i represents each value in the sample, N is the size of sample, $\bar{z}$ is the mean of the sample, and γ describes the relative stability. A lower γ indicates greater stability of the data. Table 4 illustrates the results, showing that WordBlitz has best attack stability on both the English and Chinese datasets.

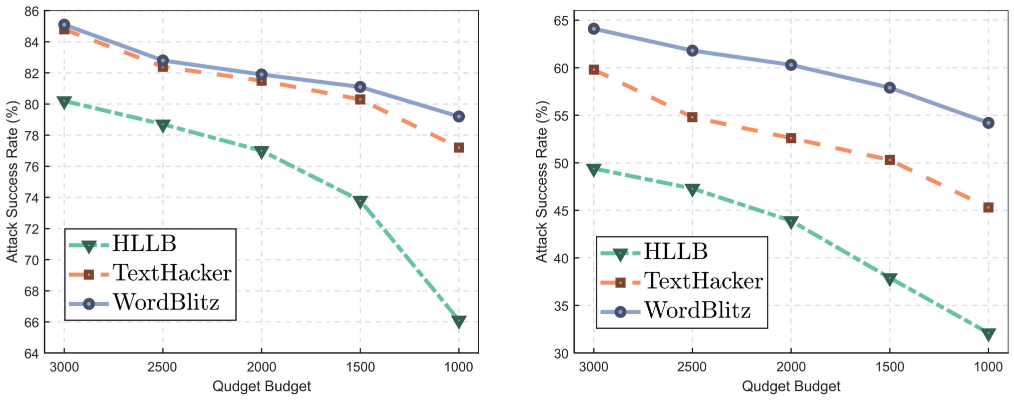
Table 4. Stability evaluation with the BERT model.

Attack	**English:Yelp**			**Chinese:Hotel**		
	γ (Succ.)	**γ (Pert.)**	**γ (Sim.)**	**γ (Succ.)**	**γ (Pert.)**	**γ (Sim.)**
HLBB	0.0195	0.0259	0.0075	0.0307	0.0298	0.0093
TextHacker	0.0092	0.0188	0.0057	0.0175	0.0274	0.0071
WordBlitz	**0.0050**	**0.0152**	**0.0042**	**0.0117**	**0.0226**	**0.0056**

The bold number denotes the best performance value for each dataset.

4.4. *Efficiency Evaluation*

In practice, the victim model can easily defend attacks based on their anomalous access frequency, which challenges the efficiency of attack methods. It is obvious that the query budget of the victim model is highly related to the efficiency. Hence, we validated the efficiency of the three methods using different query budgets, taking attacks on the IMDB and Waimai datasets with BERT model as an example. Figure 3 shows the results.



(a) Attack BERT on IMDB (English) (b) Attack BERT on Waimai (Chinese)

Figure 3. Efficiency evaluation with the BERT model.

It can be observed that the success rate of all three attack methods decreases with decreasing query budget, with the change from 1500 to 1000 being more obvious. Taking the IMDB dataset as an example, HLBB exhibits the highest deterioration in attack success rate, reaching up to 7.7%. This can be attributed to its reliance on extensive random substitutions for searching adversarial examples instead of memorizing word importance. Consequently, it is only suitable for scenarios with loose query restrictions. TextHacker experiences a lower decline of 3.1% due to its ability to remember word importance using a weight table. However, because the weight table is randomly initialized, its accuracy decreases significantly when the query budget is strictly limited. WordBlitz demonstrates the most consistent performance, with only a marginal decrease of 1.9% in attack success rate compared to the other methods. Even under the strictly limited query budget ($\leq$1000), it is able to achieves a remarkable success rate of 79.2%, which is due to the improvement in adversarial transferability with data augmentation. The experimental results on the Chinese Waimai dataset are similar, with the gap between the three methods being even more significant due to the broader search space of Chinese candidate pool, which requires high efficiency.

4.5. Ablation Study

To verify the influence of the DataAugment (DA) components in WordBlitz, we conducted an ablation study on BERT using the Waimai dataset with a query budget of 2000. We removed the DA component to validate its contribution. The results are shown in Table 5. The ablation experiment demonstrates that WordBlitz with DA has better performance than the version without DA. This shows the superiority of the DataAugment component, which learn more knowledge of the target model from history and enhances the adversarial transferability.

Table 5. Ablation study of the DataAugmentat (DA) component of WordBlitz, using the Waimai dataset on BERT.

Victim Model	Attack Method	Waimai		
		Succ.	**Pert.**	**Sim.**
BERT	WordBlitz without DA	58.1	19.2	82.5
	WordBlitz with DA	**60.3**	**18.9**	**83.8**

The bold number denotes the best performance value for each dataset.

4.6. Case Study

Table 6 shows adversarial sample cases generated by WordBlitz when attacking BERT. When attacking English texts, we used synonyms to replace the key words 'brilliant' and 'moving'. When attacking Chinese texts, we used homophones to replace the key word '香'. By perturbing a small number of key words, WordBlitz was able to successfully change the output labels.

Table 6. Case study of WordBlitz.

Language		Example	Label
English	Original	**Brilliant** and **moving** acts by Tom Courtenay and Peter Finch.	Pos
	Adversarial	Showy and emotional acts by Tom Courtenay and Peter Finch.	Neg
Chinese	Original	太香 **(xiāng, delicious)** 太大了，全是肉,嘴唇都撑裂了。	Pos
	Adversarial	太响 (xiǎng, noisy) 太大了，全是肉,嘴唇都撑裂了。	Neg

The red words denotes the replaced word. The bold words denotes the word which is used for replacement.

5. Conclusions

This paper proposes an efficient hard-label attack method called WordBlitz for generating high-quality adversarial samples with a strictly limited query budget. WordBlitz uses an Attack Parameter Set (APS) to remember word importance, which is initialized with a substitute model based on adversarial transferability. Then, adversarial examples are generated and optimized with the APS. Meanwhile, the APS is updated based on the attack results. This method overcomes the issue of low efficiency caused by random initialization. Experimental results show that WordBlitz achieves high efficiency and effectiveness, particularly on more complex languages. Compared to baselines, WordBlitz has a higher attack success rate and lower perturbation costs, especially in scenarios where query budgets are strictly limited. This means that it can be applied to real-world scenarios. In addition, the results of an ablation study prove that the DataAugment module improves the adversarial transferability.

In the future, we intend to further explore methods leveraging adversarial transferability to make our methods more general against other large-scale language models which are more robust. Ultimately, our goal is to reveal the flaws of textual classifiers in terms of adversarial robustness in order to make future models more robust. We think that there are two potential approaches. (1) In observing the generated adversarial samples, we found that their word frequency distribution and word weight distribution changed greatly. These features could be used to construct a detector to identify adversarial samples. (2) Because Chinese adversarial examples use features such as phonemes and glyphs, adding these features to train a robust model may be a potential defense. In future research, we will further explore such defense methods to improve the robustness of neural networks.

Author Contributions: Conceptualization, X.L. and H.L.; methodology, X.L.; resources, H.L. and Y.S.; writing—original draft preparation, X.L.; writing—review and editing, X.L. and H.L.; funding acquisition, H.L. and Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Nos. 62172051, and 62272052).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Nguyen, A.; Yosinski, J.; Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015 ; pp. 427–436.
2. Zhang, W.E.; Sheng, Q.Z.; Alhazmi, A.; Li, C. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol. (TIST)* **2020**, *11*, 24. [CrossRef]
3. Wang, W.; Wang, R.; Wang, L.; Wang, Z.; Ye, A. Towards a robust deep neural network in texts: A survey. *arXiv* **2019**, arXiv:1902.07285.
4. Ye, M.; Miao, C.; Wang, T.; Ma, F. TextHoaxer: Budgeted hard-label adversarial attacks on text. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 12–17 June 2022; Volume 36, pp. 3877–3884.
5. Maheshwary, R.; Maheshwary, S.; Pudi, V. Generating natural language attacks in a hard label black box setting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, pp. 13525–13533.
6. Emmery, C.; Kádár, Á.; Chrupała, G. Adversarial Stylometry in the Wild: Transferable Lexical Substitution Attacks on Author Profiling. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, 19–23 April 2021; pp. 2388–2402.
7. Zhu, Y.; Chen, Y.; Li, X.; Chen, K.; He, Y.; Tian, X.; Zheng, B.; Chen, Y.; Huang, Q. Toward understanding and boosting adversarial transferability from a distribution perspective. *IEEE Trans. Image Process.* **2022**, *31*, 6487–6501. [CrossRef] [PubMed]
8. Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; Yuille, A.L. Improving transferability of adversarial examples with input diversity. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2730–2739.
9. Li, L.; Ma, R.; Guo, Q.; Xue, X.; Qiu, X. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6193–6202.
10. Garg, S.; Ramakrishnan, G. BAE: BERT-based Adversarial Examples for Text Classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6174–6181.
11. Li, D.; Zhang, Y.; Peng, H.; Chen, L.; Brockett, C.; Sun, M.T.; Dolan, W.B. Contextualized Perturbation for Textual Adversarial Attack. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 5053–5069.
12. Maheshwary, R.; Maheshwary, S.; Pudi, V. A strong baseline for query efficient attacks in a black box setting. *arXiv* **2021**, arXiv:2109.04775.
13. Lee, D.; Moon, S.; Lee, J.; Song, H.O. Query-efficient and scalable black-box adversarial attacks on discrete sequential data via bayesian optimization. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD USA, 17–23 July 2022; pp. 12478–12497.

14. Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.J.; Srivastava, M.; Chang, K.W. Generating Natural Language Adversarial Examples. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2890–2896.
15. Zhao, Z.; Dua, D.; Singh, S. Generating Natural Adversarial Examples. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
16. Ribeiro, M.T.; Singh, S.; Guestrin, C. Semantically equivalent adversarial rules for debugging NLP models. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 856–865.
17. Yu, Z.; Wang, X.; Che, W.; He, K. TextHacker: Learning based Hybrid Local Search Algorithm for Text Hard-label Adversarial Attack. *arXiv* **2022**, arXiv:2201.08193.
18. Vijayaraghavan, P.; Roy, D. Generating black-box adversarial examples for text classifiers using a deep reinforced model. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, 16–20 September 2019; Proceedings, Part II.; Springer: Berlin/Heidelberg, Germany, 2020; pp. 711–726.
19. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into Transferable Adversarial Examples and Black-box Attacks. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
20. Wu, D.; Wang, Y.; Xia, S.T.; Bailey, J.; Ma, X. Skip Connections Matter: On the Transferability of Adversarial Examples Generated with ResNets. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
21. Lin, J.; Song, C.; He, K.; Wang, L.; Hopcroft, J.E. Nesterov Accelerated Gradient and Scale Invariance for Adversarial Attacks. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
22. Wang, X.; Ren, J.; Lin, S.; Zhu, X.; Wang, Y.; Zhang, Q. A Unified Approach to Interpreting and Boosting Adversarial Transferability. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 3–7 May 2021.
23. Wang, X.; He, K. Enhancing the transferability of adversarial attacks through variance tuning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1924–1933.
24. Eberhart, R.; Kennedy, J. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks. Citeseer, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
25. Cer, D.; Yang, Y.; Kong, S.y.; Hua, N.; Limtiaco, N.; John, R.S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; et al. Universal sentence encoder for English. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Brussels, Belgium, 31 October–4 November 2018; pp. 169–174.
26. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Toronto, ON, Canada, 2014.
27. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
28. Kenton, J.D.M.W.C.; Toutanova, L.K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 3–5 June 2019; pp. 4171–4186.
29. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Tian, H.; Wu, H.; Wang, H. Ernie 2.0: A continual pre-training framework for language understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8968–8975.
30. Zeng, A.; Liu, X.; Du, Z.; Wang, Z.; Lai, H.; Ding, M.; Yang, Z.; Xu, Y.; Zheng, W.; Xia, X.; et al. GLM-130B: An Open Bilingual Pre-trained Model. In Proceedings of the Eleventh International Conference on Learning Representations, Vienna, Austria, 25 April 2022.
31. Maas, A.; 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, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
32. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. *arXiv* **2015**, arXiv:1509.01626.
33. Pang, B.; Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Ann Arbor, MI, USA, 25 June 2005; pp. 115–124.