





## Article

# Evaluating and Enhancing the Robustness of Sustainable Neural Relationship Classifiers Using Query-Efficient Black-Box Adversarial Attacks

Ijaz Ul Haq <sup>1</sup>, Zahid Younas Khan <sup>1,2</sup>, Arshad Ahmad <sup>3</sup>, Bashir Hayat <sup>4</sup>, Asif Khan <sup>1</sup>, Ye-Eun Lee <sup>5</sup> and Ki-Il Kim <sup>5,\*</sup>

<sup>1</sup> School of Computer Science and Technology, Beijing Institute of Technology, Beijing 10081, China; engr.ijaz@outlook.com (I.U.H.); zyounask@gmail.com (Z.Y.K.); asifkhan2017@gmail.com (A.K.)

<sup>2</sup> Department of Computer Science and Information Technology, University of Azad Jammu and Kashmir, Muzaffarabad 13100, Pakistan

<sup>3</sup> Department of IT and Computer Science Pak-Austria Fachhochschule Institute of Applied Sciences and Technology, Haripur 22620, Pakistan; yaarshad@gmail.com

<sup>4</sup> Institute of Management Sciences Peshawar, Peshawar 25100, Pakistan; bashir.hayat@imsciences.edu.pk

<sup>5</sup> Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, Korea; yeyettt@naver.com

\* Correspondence: kikim@cnu.ac.kr



**Citation:** Haq, I.U.; Khan, Z.Y.; Ahmad, A.; Hayat, B.; Khan, A.; Lee, Y.-E.; Kim, K.I. Evaluating and Enhancing the Robustness of Sustainable Neural Relationship Classifiers Using Query-Efficient Black-Box Adversarial Attacks. *Sustainability* **2021**, *13*, 5892. <https://doi.org/10.3390/su13115892>

Academic Editor: Manuel Fernandez-Veiga

Received: 7 April 2021

Accepted: 18 May 2021

Published: 24 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Abstract:** Neural relation extraction (NRE) models are the backbone of various machine learning tasks, including knowledge base enrichment, information extraction, and document summarization. Despite the vast popularity of these models, their vulnerabilities remain unknown; this is of high concern given their growing use in security-sensitive applications such as question answering and machine translation in the aspects of sustainability. In this study, we demonstrate that NRE models are inherently vulnerable to adversarially crafted text that contains imperceptible modifications of the original but can mislead the target NRE model. Specifically, we propose a novel sustainable term frequency-inverse document frequency (TFIDF) based black-box adversarial attack to evaluate the robustness of state-of-the-art CNN, CGN, LSTM, and BERT-based models on two benchmark RE datasets. Compared with white-box adversarial attacks, black-box attacks impose further constraints on the query budget; thus, efficient black-box attacks remain an open problem. By applying TFIDF to the correctly classified sentences of each class label in the test set, the proposed query-efficient method achieves a reduction of up to 70% in the number of queries to the target model for identifying important text items. Based on these items, we design both character- and word-level perturbations to generate adversarial examples. The proposed attack successfully reduces the accuracy of six representative models from an average F1 score of 80% to below 20%. The generated adversarial examples were evaluated by humans and are considered semantically similar. Moreover, we discuss defense strategies that mitigate such attacks, and the potential countermeasures that could be deployed in order to improve sustainability of the proposed scheme.

**Keywords:** robust; sustainability; adversarial attack; black-box attack; TFIDF; relation extraction; deep neural networks

## 1. Introduction

Relation extraction (RE) is the classification of relations between two entities. It is important and useful in several applications of natural language processing (NLP) [1], such as question answering [2], information extraction [3], knowledge-base machine translation, and document summarization. Recently, deep neural networks (DNNs) have been applied successfully in a variety of NLP tasks [4,5]. The outstanding performance of these DNNs on complex data has received attention from researchers and are now frequently used in different domains such as NLP, computer vision, speech recognition, etc. Among these

neural models, RNNs, CNNs, GNNs, and pre-trained models such as BERT have obtained state-of-the-art results for RE. RNNs look for a sequence of words in a text to capture dependencies of words and text structures. CNNs recognize the key phrases of text using pattern recognition and feature extraction techniques, while GNNs capture the internal structures of the graphs such as semantic and syntactic parse trees. Pre-trained models such as BERT, which use transfer learning techniques, are pre-trained on huge text corpus, e.g., the Wikipedia text corpus, and after fine tuning, they can be applied to many different kinds of NLP tasks. In particular, complex deep neural RE models have exhibited higher accuracy on supervised RE datasets, such as the SemEval-2010 Task 8 [6] and the TACRED dataset [7]. Traditional RE methods, such as kernel- and feature-based methods, suffer from erroneous labeling [8].

To overcome this, supervised DNN models have been used [9]. However, the characterization understanding of the behavior of these complex neural models is a challenging task. The statistical learning perspective implies that more complex models are more prone to brittleness [10]. There has been an ongoing discussion regarding the sustainability of these models as to the extent to which they understand natural language [11] and utilize the cues and unintentional biases in the training dataset [12,13]. The behavior of these models becomes unpredictable if they are evaluated or tested on data outside the defined distribution of the training dataset; therefore, to expose these “blind spots,” adversarial attacks have been used for robustness evaluation of these deep machine learning models. Normally, the attacker applies either a white-box attack or black-box attack. In a white-box attack on textual data, the attacker uses the gradient information of the victim model to find the positions of important words and perturb them in multiple ways. On the other hand, black-box attacks are blind attacks, in which the attacker has no access to gradients of the victim model. More often, adversarial text generation in a black-box attack is done by finding the significant words that has a high impact on the confidence score of the victim model and then perturbing those words. Supervised RE models currently use a test set for evaluation and measurement. Higher accuracy scores on test sets indicate that the model is effective only if the real world is represented by the test set [14]. However, the distribution of test and training sets are most likely to be the same, as they are generated in parallel and do not necessarily represent real-world scenarios [15].

In this study, we use focus on evaluating and improving the robustness of supervised RE models in the aspects of sustainability under black-box adversarial attack settings. These RE models are responsible for predicting the class of relations between two mentioned entities. RE is a multiclass classification task. Accordingly, each supervised RE dataset is designed to classify different classes of relations. As sensitive automated systems such as question answering use RE models at the back-end, it is meaningful to fully understand the extent to which these models are sustainable or being affected by adversarial attacks and their degree of robustness to wording changes. Furthermore, it is also necessary to determine the deficiencies of the datasets used to train these models [10].

### 1.1. Challenges

As adversarial attacks have been successful in image and speech classification [16,17], researchers have attempted to extend them to tasks related to NLP, such as text classification, sentiment analysis, machine translation, machine comprehension, and text entrainment. Generating adversarial examples for text domains remains challenging because of the discrete nature of textual data. Furthermore, in addition to the ability to mislead the text classifier, it is important to preserve the utility of the original text after the attack:

1. Human prediction should remain unchanged.
2. Semantic similarity should be maintained.
3. Adversarial examples should appear natural and fluent.

Previous studies have barely fulfilled all these requirements. For example, in [18] single-word erasure was used, and in [11], non-related phrases were added and removed. These types of changes resulted in unnatural text. Moreover, under black-box settings,

it is difficult to identify significant text items for classification. Black-box adversarial attacks are evaluated based on the number of queries made to identify significant items for classification. It is highly important to consider the cost associated with the number of such queries in attacks on real-world systems. Attack time can be minimized by reducing the number of queries [19,20]. A method is considered more efficient if the average clock time for perturbing a single example is shorter. Query-based approaches have primarily been used to calculate word importance by deleting a certain word in a sentence, and to determine whether the predictions have changed. This method is effective, but it invokes the classifier for every word, which is time-consuming; thus, the following questions arise in the aspects of sustainability:

1. Determining significant words for an adversarial attack and invoking the classifier as rarely as possible.
2. Generating adversarial examples that can mislead the classifier considering the characteristics of RE datasets and models.
3. Evaluating the success of an attack.
4. Improving the robustness of these models.

### 1.2. Contributions

To better understand the robustness of supervised NRE models, we propose certain sustainable adversarial attack methods and accordingly generate adversarial examples to address the aforementioned questions. As mentioned earlier, the test sets have almost the same distribution as the training datasets. We used term frequency—inverse document frequency (TFIDF) to determine important words in sentences of each class label from the test set. This method, which has not previously been used, does not invoke the classifier to repeatedly determine significant words. It is conceivable that test sets contain hints for attackers because by obtaining the TFIDF of all correctly classified sentences of a specific class in a test set, TFIDF can identify up to an average of 70% of important words for classification. To perturb a word, we used both character-level attacks (CLAs) and word-level attacks (WLAs) to fully understand the vulnerabilities of NRE models.

We apply the proposed framework to attack state-of-the-art and representative NRE models, namely, CNN [21], attention Bi-LSTM [22], and R-Bert [23] for the SemEval-2010 Task 8 dataset, and PA-LSTM [7], C-GCN+PA-LSTM [24], and SpanBERT [25] for the TACRED RE dataset. We automatically evaluated attack success by comparing the classification accuracy before and after the attack with variants of our attack and modified baseline algorithms of other text classification tasks; we also used human evaluations. The proposed adversarial attack successfully reduced the accuracy of the targeted models to under 20%, and not more than 20% of the words were perturbed in a sentence. The adversarial sentences were correctly classified by the human judges. At the end of the adversarial attack, we performed adversarial training to retrain the model for improved sustainability and robustness. To the best of our knowledge, this is the first study to evaluate and measure the robustness of supervised RE models under adversarial examples. The contributions of this study can be summarized as follows:

1. We propose a novel query-efficient TFIDF-based black-box adversarial attack and generate semantically similar and plausible adversarial examples for NRE task.
2. Our mechanism evaluates supervised RE models using black-box adversarial attacks; this has not been previously undertaken. It was demonstrated that no available open-source RE model is robust and sustainable to character- and word-level perturbations.
3. Our proposed adversarial attack makes use of test samples to find significant words in a sentence, therefore reducing the number of queries and time required to generate an adversarial example.
4. In comparison to other similar black-box attacks on text classification and entailment tasks with a minor modification in their algorithms for RE dataset (constraints on modifying the mentioned entities). Our method achieves a higher attack success rate in the lowest number of queries.

5. We further discussed two potential defense mechanisms to defend the models against the aforementioned attacks along with evaluations.

The rest of the paper is organized as follows. Related work is briefly reviewed in Section 2. The proposed method is described in Section 3. Experiments on datasets and their corresponding target models are presented in Section 4. Section 5 describes the effectiveness and efficiency of the proposed attack model. Attack evaluation is carried out in Section 6. The transferability of the generated adversarial sentences between models is discussed in Section 7. Defense strategies against adversarial attacks on supervised RE models are considered in Section 8. The paper is concluded in Section 9.

## 2. Related Work

The evaluation of DNN-based classifiers has recently attracted considerable attention. Researchers have crafted unperceivable adversarial perturbations that can mislead such classifiers. These adversarial examples were first introduced in [26] to evaluate the robustness of state-of-the-art CNN-based image classifiers using small perturbations applied to input images. It was demonstrated that the classifier was vulnerable to such perturbations.

It is impractical and expensive to generate adversarial examples [26]; therefore, various generation methods have been proposed, including [16,27] gradient [28], decision function, and [29] evolution-based methods. As adversarial attacks were first used in computer vision and were later adopted in NLP, few attempts have been made to attack NLP neural models and evaluate their robustness. Unlike in the case of images, where the embedding space is continuous, and perturbations are carried out by altering pixels, adversarial attacks cannot be directly made on textual data because of the discrete nature of these data. In [11,30], difficulties were reported in using a fast gradient sign method (FGSM) [16] to attack an RNN-based neural NLP model owing to the intrinsic differences between textual data and images.

In [27], TextFool also used FGSM to determine significant text items in the context of text classification, and three attack strategies, namely, addition, removal, and modification, were proposed to evaluate a CNN text classifier [31]. However, these techniques were manually performed.

TextFool was modified in [32]. An adverb  $w_i$  that contributes more to the text classification task is first removed. Then, it is checked whether the grammar is incorrect, and in this case, a word  $p_j$  is inserted. This word is selected from a candidate pool consisting of typos, synonyms, and genre-specific words. If the grammar or similarity is not satisfied, then the word  $w_i$  is replaced with  $p_j$ . It has been demonstrated that this method is more effective than TextFool because the POS and grammar of the original text are not affected.

In [33,34], character-level perturbations called Hotflip were performed by inserting, deleting, and swapping letters, causing translation errors and wrong outputs in text classification and machine translation tasks. In [35], this was extended by adding a controlled attack in which a specific word is removed from the output, and a targeted attack in which a specific word is replaced. These methods change the meaning of sentences and intentionally introduce errors. Reference [36,37] discussed techniques of adding random noise to text by changing the tokens of word, while [38] proposed word dropout strategy to improve the sustainability of language models. This does not preserve the semantics of the sentence and modify the sentence in an unnatural way.

Small manually constructed adversarial datasets were used to evaluate the vulnerability of sentiment analysis [39,40], machine translation [41,42], and natural language understanding [43] systems. In this type of attack, the attacker would be highly confident regarding the labels and grammatical correctness of instances; however, it is expensive to exploit human knowledge of language and is difficult to construct and scale larger datasets. Recently, in [44], an adversarial dataset was introduced for stress test evaluation in natural language inference. This dataset contains methods for generating new adversarial claims to evaluate the sustainability of state-of-the-art models in the context of a shared task called MultiNLI by using three types of perturbations: meaning-altering transformation,

rule-based transformation, and distractor phrases. This approach is comparatively cheaper than manual constructing a dataset, as several instances can be changed by one rule.

The problem of query efficiency of black-box adversarial attacks has been addressed by many researchers [45–47] in image classification domain but it has been neglected in studies of black-box adversarial attack on textual data. Most of the textual black-box adversarial attacks are carried out by querying the victim models many times for perturbation of just one word, which invokes the model's classifier thousands of time, therefore taking more time and using more processing power.

However, in this study we focus on the query efficiency of our proposed black-box adversarial attack. In addition, the robustness issue of sustainable supervised RE models has also not been considered, and to the best of our knowledge, this is the first study to evaluate the robustness of deep neural networks for RE. Furthermore, we compare the proposed TFIDF method with previous methods of determining significance words in sentences, and evaluate the models using different character-level and word-level perturbations.

### 3. Methodology

#### 3.1. Problem Formulation

Given a set of  $N$  labels  $Y = \{y_1, y_2, y_3 \dots y_N\}$  and a set of  $N$  sentences  $S = \{s_1, s_2, s_3 \dots s_N\}$ , each sentence has two mentioned entities  $(e_1, e_2)$  and  $n$  words  $w_i = \{w_1, w_2, e_1 \dots e_2 \dots w_n\}$ . We have a pre-trained relation classification model  $F : S \rightarrow Y$  that maps features from the input text space  $X$  to the feature space of  $N$  labels  $Y$ . The aim of the adversary is to generate valid adversarial examples  $S_{adv}$  for a sentence  $s_i \in S$  such that  $F(S) = y$  and  $F(S_{adv}) = z$ , with  $y \neq z$ . The adversarial example should also be semantically and syntactically similar to the original sentence, that is,  $Sim(S, S_{adv}) \geq \epsilon$ , where  $\epsilon$  is a threshold value between 0 and 1 indicating the minimum semantic similarity between  $S$  and  $S_{adv}$ . Therefore, the similarity function is  $Sim : X \times X \rightarrow (0, 1)$ .

#### 3.2. Threat Model

Compared with the case of a white-box attack, in which the attacker has complete knowledge of the model, in a black-box setting, attackers have no access to the target model architecture, training data, or parameters. Therefore, in this case, the attacker can only query the target model and obtain a confidence score or prediction. The black-box attack should meet the following three requirements:

1. It should generate fluent adversarial examples that are semantically similar to the original sentence.
2. The target NRE model/classifier should be invoked as few times as possible.
3. It should fool the NRE model into producing erroneous outputs.

#### 3.3. Adversarial Attack

We intend to experiment with different variants of the attack model. The adversarial attack consists of the following steps:

1. Determining important words and sorting them in descending order according to their importance.
2. Using these words to generate adversarial sentences.
3. Checking the similarity constraint between the original and adversarial sentences.
4. Checking whether the adversarial sentence changes the output of the model.

We propose three methods for determining the importance of a word: TFIDF-based, query-based (QB), and a combination thereof. We generate two types of perturbations in a sentence: word-level and character-level. We combine these techniques as follows:

- TFIDF+(WLA/CLA).
- QB+(WLA/CLA).
- Combined (TFIDF-QB+(WLA/CLA)).



We describe each step of these methods in detail in the next section, and the results from different variants are presented in Section 6. The proposed method for generating adversarial sentences and misleading supervised RE models is shown in Algorithm 1. The algorithm shows the third variant of the proposed attack model, that is, combined (TFIDF-QB+(WLA/CLA)), as it covers all types of the proposed attack methods.

### 3.3.1. Step 1: Word Importance Ranking (Lines 1–8 and 24–28)

The first step of a black-box attack is to determine important words. In supervised relation classification, each sentence is given with two mentioned entities. The objective is to determine important words around these entities, for example, those words that come before, after, or between the entities. In the process for generating adversarial examples, these entities are not changed or modified. We propose three methods and compare their performance in terms of the invocation frequency of the classifier. The following methods are used to select important words.

1. TFIDF-based word importance ranking (TFIDF-WIR).
2. QB word importance ranking (QB-WIR).
3. Combined TFIDF and QB word importance ranking ((TFIDF+QB)-WIR).

#### TFIDF-Based Word Importance Ranking

Term frequency-inverse document frequency provides statistics regarding word importance in a document, corpus, or collection. For example, in our case, if we take the SemEval-2010 Task 8 dataset as an example, we are given nine types of relations, and a label “other” if there is no relation identified. We generated nine documents, one for each class label from the test set. Each document contains the sentences correctly classified by all target models for that specific class label. An example of the cause–effect relationship from the SemEval-2010 Task 8 dataset is as follows:

Traffic < *e1* > vibrations < /*e1* > on the street outside caused the < *e2* > movement < /*e2* > of the light.

It has been demonstrated that not all words contribute equally to the classification of semantic relations [48]. To determine the relation, the word “caused” is of particular significance, whereas the word “street” is less correlated with the semantics of cause–effect relationships. The TFIDF list of important words of each class-label document provides almost 70% of the important words and their weights, which are significant for the classification of a relation. Figure 1 shows the lists of words belonging to particular classes. The words in each document are ranked in descending order of their weights. The mentioned entities of the sentences are not included in these lists because they should not be changed, as the goal of RE is to determine the relation between the entities. After obtaining *TFIDF*, we arrange the words in each document according to their importance ranking. We further use the NLTK library to filter out a few non-related stop words such as “the,” “when,” and “none,” but not words such as “inside,” “by,” and “from,” as some stop words may also affect the prediction output. This method does not invoke the classifier because it checks the target word in the generated list of important words.

**Algorithm 1:** Black-box TFIDF-QB combined attack.

---

**Input:** Sentence  $S = \{w_1, w_2 \dots e_1 \dots e_2 \dots w_n\}$ , original label  $y$ , classifier  $F$ ,  $Sim()$  Function, threshold  $\epsilon$ , and  $TFIDF[y_{list}]$

**Output:**  $S_{adv}$  adversarial sentence

```

1 Initialize:  $S_{adv} \leftarrow S$ 
2 for each word  $w_i$  in  $S$  do
3   if  $w_i \neq e_1$  or  $e_2$  (mentioned entities) then
4     for each word  $k$  in  $TFIDF[y_{list}]$  do
5       if  $k = w_i$  If the word is present in  $TFIDF[y_{list}]$  then
6          $Z_{list} =$  Get the importance score of top 4  $w_i$ 
7 if  $Z_{list} \neq null$  then
8    $Z_{ordered} \leftarrow$  Sort  $Z_{list}$  according to importance Score in  $TFIDF[y_{list}]$ 
9   Input: Attack type
10  if Attack type = "CharAttack" then
11    for  $w_j$  in  $Z_{ordered}$  do
12       $S_{adv} = CharAttack(w_j, S_{adv}, y, F(.))$ 
13      if  $Sim(S, S_{adv}) \leq \epsilon$  then
14        Return none
15      else if  $F(S_{adv}) \neq y$  then
16        Attack successful, Return  $S_{adv}$ 
17  else if Attack type = "WordAttack" then
18    for  $w_j$  in  $Z_{ordered}$  do
19       $S_{adv} = WordAttack(w_j, S_{adv}, y, F(.))$ 
20      if  $Sim(S, S_{adv}) \leq \epsilon$  then
21        Return none
22      else if  $F(S_{adv}) \neq y$  then
23        Attack successful, Return  $S_{adv}$ 
24 else if  $Z = null$ , that is, no word found in  $TFIDF[y_{list}]$  then
25 for each word  $w_i$  in  $S$  do
26   if  $w_i \neq e_1$  or  $e_2$  (mentioned entities) then
27      $Z_{list} =$  Compute the word importance  $P(w_{i,y})$  from Equation (4)
28  $Z_{ordered} \leftarrow$  Sort  $Z_{list}$  according to  $P(w_{i,y})$ 
29 Repeat steps from lines 9–23

```

---

<b>Cause-effect</b> 1. Caused 2. From 3. Resulted 4. Triggered . . . . 1846. Small 1847. Long	<b>Component-Whole</b> 1. Of 2. Comprises 3. Contains 4. From . . . . 1812. Way 1813. Set	<b>Content-Container</b> 1. In 2. Inside 3. Hidden 4. Full . . . . 936. Get 937. Little	<b>Entity-Destination</b> 1. Poured 2. Put 3. Arrived 4. Sent . . . . 876. Made 877. Also	<b>Entity-Origin</b> 1. From 2. Away 3. Originated 4. Departed . . . . 1380. Years 1381. Set
<b>Instrument-Agency</b> 1. Using 2. Tool 3. Help 4. Exam . . . . 875. Found 876. Still	<b>Member-Collection</b> 1. Member 2. Of 3. Various 4. Like . . . . 1833. Since 1834. Another	<b>Message-Topic</b> 1. Subject 2. Topic 3. Related 4. Informed . . . . 1234. Little 1235. Long	<b>Product-Producer</b> 1. Produced 2. Built 3. Maker 4. Constructed . . . . 1547. Sentences 1548. Since	

**Figure 1.** TFIDF word importance list for the SemEval-2010 Task 8 test set.

**TF** is the number of times a word appears in a document divided by the total number of words in the document:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}. \quad (1)$$

**IDF** measures the weight of a term across all the documents, and it is calculated as follows:

$$idf(w) = \log\left(\frac{N}{df_t}\right). \quad (2)$$

**TF-IDF:** The TF-IDF score is obtained by multiplying TF and IDF:

$$TF - IDF = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \quad (3)$$

### Query-Based Word Importance Ranking

This technique has been used in several studies [49,50] to determine the importance of a word in black-box settings. As RE is different from other classification tasks, we adopted this method with a minor change, that is, the mentioned entities are not included in the word importance ranking. Given the sentence  $s$  consisting of  $n$  words  $w_i$ , to determine the words that are significant for the classification of the relation between the two mentioned entities, the words around the mentioned entities should be individually deleted so that their significance may be checked. If the deletion of a specific word reduces the confidence or the prediction score of the original label, we consider it an important word. Thereby, the words are ranked according to the change in the confidence score of the original label. This method invokes the classifier to determine the importance of each word in a sentence. To represent the sentence, we have three scenarios: a word may be deleted before, between, and after the mentioned entities. Thus, the sentence after the removal of the  $i_{th}$  word can be as follows:

$$\begin{aligned} \bar{S} &= \{w_1 \dots w_{i-1}, w_{i+1} \dots e_1 \dots e_2 \dots w_n\}, \\ \bar{S} &= \{w_1 \dots e_1 \dots w_{i-1}, w_{i+1} \dots e_2 \dots w_n\}, \text{ and} \\ \bar{S} &= \{w_1 \dots e_1 \dots e_2 \dots w_{i-1}, w_{i+1} \dots w_n\}. \end{aligned}$$

Hence, the importance score of  $w_i$  is



$$P(w_{i,y}) = Fy(S) - Fy(\bar{S}). \quad (4)$$

$Fy(S)$  is the classification accuracy of the original sentence, and  $Fy(\bar{S})$  is the classification accuracy after the removal of the  $i$ th word.

### Combined TFIDF-WIR and QB-WIR

We now combine the aforementioned methods (TFIDF-WIR and QB-WIR). We assume that it is possible that not all important words may be present in the list of important words generated by TFIDF when the test set is used. It is most likely that TFIDF will determine most of the important words as it is applied to the test set, the distribution of which is almost the same as that of the training set. However, if it fails, the QB method will be used. Algorithm 1 shows the combined method for selecting important words from lines 1–8 (TFIDF-WIR) and 24–28 (QB-WIR).

#### 3.3.2. Step 2: Word Transformer (Line 10–23):

After the selection of important words in Step 1, we propose character- and word-level perturbations. Line 9 of Algorithm 1 requests the perturbation type according to which adversarial sentences are to be generated, and lines 12 and 19 call functions corresponding to character- and word-level perturbations, respectively. These functions are described in Algorithms 2 and 3.

#### Character-Level Attack

This type of perturbation is used to modify the characters of important words. It has been demonstrated that the meaning of a sentence is inferred or preserved by a human reader if a small number of characters are changed [33,45]. Moreover, character-level perturbations can strongly mislead the prediction process in other text classification tasks [43]; however, its effect on relation classification remains unknown. The characters of the selected important words can be perturbed in various manners. Algorithm 2 shows the function for CLA. This function provides five adversarial sentences according to five suggested CLAs described below. Subsequently, the best sentence with the lowest prediction score of the original label  $y$  is selected.

---

#### Algorithm 2: CLA.

---

```

1 Function CharAttack( $w, x, yF(\cdot)$ )
2  $\bar{X} = \text{GenerateAdv}(w, x)$ 
3 for  $\bar{x}_k$  in  $\bar{X}$  do
4    $\text{Score}(k) = Fy(x) - Fy(\bar{x}_k)$ 
5  $\text{Adv}_{\text{best}} = \arg \max_{\bar{x}_k} \text{Score}(k)$ 
6 Return  $\text{Adv}_{\text{best}}$ 
7 end Function

```

---

The function  $\text{GenerateAdv}(w, x)$  in Algorithm 2 is important because it applies all types of character-level perturbations to the given word  $w$  in the sentence  $x$ . We use the following types of character-level perturbations: Insert-C, Repeat-C, Swap-C, Delete-C, and Replace-C. These perturbations are described below, and they are applied to the letters between the first and last character of the word.

1. **Insert-C:** The inserting strategy can be applied in several ways. For example, there are  $26^n$  combinations for inserting a character from the latin alphabet. Special characters such as “!” and “@”, as well as a space “ ” between the first and the last character of a word can also be inserted. We opt to insert a symbolic character because the resulting mistyped words can be easily understood by humans; however, the embedding vectors are different from those of the original words, and thus the classifier can be fooled.

2. **Repeat-C:** This is almost the same concept as the inserting technique, but here we select a random character between the first and last character of an important word and repeat it once. As in the previous case, the resulting word can be easily identified by human readers as the original.
3. **Swap-C:** This randomly swaps two adjacent letters without changing the first and last letter.
4. **Delete-C:** This deletes a letter between the first and last letter. This perturbation can also be easily recognized by a human, but the classifier may fail.
5. **Replace-C:** This replace the original letters with visually similar letters. For example, “o” can be replaced with zero “0,” the letter “a” can be replaced with “@,” or lower case letters can be replaced with upper case.

### Word-Level Attack

This type of perturbation has previously been used differently [44,46–48]. We propose three types of WLAs: synonym replacement, word swap, and word repetition. The function for MLAs is presented in Algorithm 3. It produces adversarial sentences with all three types of word-level perturbations. Subsequently, we select the adversary that yields the best result in terms of the prediction reduction of the original class label.  $GenerateAdv(w, x)$  has the same function as in a CLA, but here, it returns word-level perturbed adversarial sentences. Each type of word-level perturbation is described in detail below.

---

#### Algorithm 3: WLA.

---

```

1 Function WordAttack( $w, x, yF(.)$ )
2  $\bar{X} = GenerateAdv(w, x)$ 
3 for  $\bar{x}_k$  in  $\bar{X}$  do
4    $Score(k) = F_y(x) - F_y(\bar{x}_k)$ 
5  $Adv_{best} = \arg \max_{\bar{x}_k} Score(k)$ 
6 Return  $Adv_{best}$ 
7 end Function

```

---

1. **Synonym replacement:** This is a popular attack strategy because it preserves word semantics. This function replaces a word with a synonym. We obtained important words by the aforementioned methods and gathered a synonym list for replacement (except for the mentioned entities and a few unimportant stop words). The synonym list is initiated by the N-nearest neighbor synonyms of an important word according to the cosine similarity between words in the vocabulary. There are several ways to obtain synonyms from available resources. To represent the words, one can use “NLTK Wordnet” to obtain the top synonyms or the new counter-fitting method in [51]. In this study, we used the counter-fitting embedding space to find the synonyms of the words, because it produced the best results on the SimLex-999 dataset [52]. This dataset was designed to measure model performance in determining the semantic similarity between words. We select the top  $k$  synonyms with a distance to the selected word greater than  $\delta$  (sigma).
2. **Word swap:** This method is easy to apply, as it randomly swaps a selected important word with preceding or succeeding words. If the model is unaware of the word order, it can yield erroneous outputs. Human readers can easily understand and identify the word order in a sentence, but the classifier may fail. This perturbation function returns a modified sentence with an important word swapped with the word before or after it.
3. **Word repetition:** We can perturb a sentence by repeating some words (except for the mentioned entities important words, and a few stop words such as “the” and “is”), as repeating an important word can increase the confidence score of the original label. Therefore, this function returns a sentence with the least important words repeated.

### 3.3.3. Step 3: Semantic Similarity

There are four popular methods for evaluating the utility of the generated adversarial text: edit distance, Jaccard similarity coefficient, semantic similarity, and Euclidean distance.

Edit distance and the Jaccard similarity coefficient can be applied to raw text, whereas semantic similarity and the Euclidean distance are applied to word vectors. We use cosine semantic similarity because the other metrics can only reflect the magnitude of the generated adversarial text, and they cannot ensure that the generated perturbation preserves the semantic similarity from the original text. In this study, we use the universal sentence encoder [53], which produces high-dimensional vectors of the adversarial and the original sentence. Using these vectors, we can determine the cosine similarity between these sentences. Given two high-dimensional vectors  $P$  and  $Q$  representing the original and the adversarial sentence, respectively, the cosine similarity is defined as follows:

$$\text{Sim}(p, q) = \frac{p \cdot q}{\|p\| \cdot \|q\|} = \frac{\sum_{i=1}^n p_i \times q_i}{\sqrt{\sum_{i=1}^n (p_i)^2} \times \sqrt{\sum_{i=1}^n (q_i)^2}}. \quad (5)$$

After obtaining the cosine semantic similarity value of the original and the adversarial sentence, we verify whether the value is below or above the threshold  $\epsilon$  to control the semantic similarity of adversarial sentences.

## 4. Experiments

### 4.1. Datasets

There are number of datasets available for relation extraction tasks, the main differences in all these datasets are the type of relation that needs to be extracted and the length of the sentences. We selected SemEval 2010 Task 8 [6] and KBP TACRED [7] for our experiment. These two dataset are among the top benchmark datasets used for supervised RE. In addition, most of the work published on RE used these dataset together for experimentation. Furthermore, uniqueness of entity types, i.e., (SemEval Task 8 2010 relation between nominals, e.g., cause–effect and TACRED: Relation between subjects and objects, e.g., Person–Title) and their classification requirements (SemEval 2010 Task 8 is a multi-way classification tasks, whereas TACRED is a one-way classification task) make them the best choice to evaluate the sustainability of NRE models. We believe that these two datasets cover the most common characteristics of almost all the supervised RE datasets and are therefore well suited for our experiment. The statistics of these datasets are provided in Table 1.

**Table 1.** Dataset statistics.

Task	Dataset	Train	Dev	Test	Classes	Avg
Relation extraction	SemEval-2010 Task 8	8000	0	2717	9	19.1
	TACRED	68,124	22,631	15,509	42	36.2

### 4.2. Targeted Models

We trained three open-source state-of-the-art target models on each dataset: CNN [21], attention Bi-LSTM [22], and R-BERT [23] on SemEval-2010 Task 8, and PA-LSTM [7], C-GCN + PA-LSTM [24], and SpanBert [25] on TACRED. The codes for these models are available on GitHub. The models were trained using the same hyper-parameters as mentioned in their research papers and achieved almost the same results. The original accuracy of these models on the corresponding datasets is provided in Table 2.

**Table 2.** Original accuracy of target models before adversarial attack.

SemEval-2010Task 8	CNN	Attention Bi-LSTM	R-Bert
Original accuracy	82.7%	84.0%	89.25%
TACRED	PA-LSTM	C-GCN+PA-LSTM	SpanBert
Original accuracy	65.1%	68.2%	70.8%

### 5. Attack Efficiency and Effectiveness

To evaluate the effectiveness and efficiency of the proposed attack model, we selected 500 and 1000 sentences that were correctly classified by all the target NRE models from the test sets of SemEval-2010 Task 8 and TACRED, respectively. For SemEval-2010 Task 8, the objective of the attacker is to change the prediction label of the original sentence to one of the remaining eight labels, and for TACRED, to one of the remaining 41 labels. The attacker is not allowed to make changes to any mentioned entity, subject, or object in both tasks, but can only perturb the words around them to generate an adversarial sentence. The results of the main black-box attack TFIDF+QB-CLA and TFIDF+QB-WLA are shown in Table 3. The success of the attack is calculated by the difference in the accuracy of the original label before and after the attack. It can be seen that the proposed attack can achieve a particularly high success rate. Even though in both supervised RE tasks, less than 20% of the words were perturbed, the accuracy of the original label dropped to below 20% on average. This demonstrates that the proposed attack can always reduce prediction accuracy, regardless of the sentence length and model accuracy. Furthermore, BERT is slightly more robust than the other models on both tasks. On SemEval-2010 Task 8, the accuracy of BERT dropped from 87.7% to 22.5% after a CLA, and to 20.6% after a WLA, whereas the corresponding results for CNN and attention Bi-LSTM were 14.6% and 17.1% for CLA, and 15.2% and 17.4% for WLA, respectively. On TACRED, the accuracy of the original label in SpanBert was reduced from 69.1% to 18.4% and 23.1% by CLA and WLA, respectively. The corresponding results for PA-LSTM and C-GCN+PA-LSTM were 12.9% and 14.1%, and 13.1% and 18.7%, respectively. It can be concluded that even though BERT is more robust than CNN, GCN, and LSTM-based models, it can be fooled by adversarial attacks.

**Table 3.** Accuracy of target models after adversarial attack.

Dataset	SemEval-2010 Task 8			TACRED		
Model	CNN	Att- Bi-LSTM	R-Bert	PA-LSTM	C-GCN+PA-LSTM	SpanBert
Original	81.2%	83.4%	87.7%	62.6%	64.5%	69.1%
TFIDF+QB-CLA	14.6%	17.1%	<b>22.5%</b>	12.9%	13.1%	18.4%
TFIDF+QB-WLA	15.2%	17.4%	20.6%	14.1%	18.7%	<b>23.1%</b>
Semantic similarity (Avg)	0.81%	0.70%	0.67%	0.86%	0.81%	0.76%
Perturbed words (Avg)	9.3%	11.1%	14.2%	13.7%	14.9%	18.2%
Avg length	19.1			36.2		

It can also be observed that the average number of perturbed words and the average semantic similarity for both WLAs and CLAs are correlated. If the percentage of perturbed words is high, the semantic similarity decreases. For example, on both tasks, the word perturbation percentage in CNN is lower than in attention Bi-LSTM and BERT-based models, and therefore, the semantic similarity is higher. The word perturbation rate in BERT-based models is higher than that in all other models, demonstrating the high robustness of the former.

## 6. Attack Evaluation

As this is the first study on adversarial attacks in supervised RE, the robustness of encoders such as CNN, LSTM, GCN, and BERT has not been evaluated for this task. We applied six different methods to generate adversarial sentences by using different combinations for the selection of significant words. By comparing the results of each attack, it was demonstrated that even though in black-box settings, most previous studies on adversarial attacks in different text classification tasks used the QB approach to determine the important words, these words can be obtained from the test sets by using *TFIDF*. This is because for the classification a specific relation such as “entity-origin” from SemEval-2010 Task 8 and “Person-City\_of\_birth” from TACRED, words such as “arrive” and “died” are frequently repeated in sentences related to their class labels. These words are not important for other types of relations. Applying *TFIDF* to all groups of sentences belonging to a particular class label can provide almost 70% of the important words from each class label group. We explain this in Section 3.3.1. In the QB technique for determining word importance, the classifier is queried and invoked to determine the importance of every word. For example, for 500 sentences, the classifier is invoked nearly 1200–1500 times for SemEval-2010 Task 8, and for 1000 sentences from TACRED, it is invoked up to 2500–3000 times to determine three important words per sentence on average. By contrast, the *TFIDF* method never invokes the classifier.

Tables 4 and 5 show different aspects of the proposed attack. Column *Invok# ( $WI_m$ )* indicates the number of times that the classifier was invoked for each task to determine important words. In the *TFIDF*-based method, the classifier was not invoked at all ults, whereas in the combined method *TFIDF-QB*, the classifier was invoked 300–400 times for SemEval-2010 Task 8 and 750–800 times for TACRED; this is preferable to only using the QB-based method, the attack success rate of which is also high.

**Table 4.** Results of adversarial attack on 500 sentences of SemEval-2010 Task 8 and corresponding targeted models.

Attack Types	Victim Models	Invok# ( $WI_m$ )	AvgTime (s)	Attack Success
TFIDF-CLA	CNN	0	2.51	83.4%
	Att- Bi-LSTM	0	2.36	74.8%
	R-Bert	0	2.74	77.2%
TFIDF-WLA	CNN	0	2.42	76.4%
	Att- Bi-LSTM	0	2.21	71.5%
	R-Bert	0	2.66	73.9%
QB-CLA	CNN	1215.4	24.71	85.4%
	Att- Bi-LSTM	1299.6	25.77	83.3%
	R-Bert	1486.7	27.2	81.2%
QB-WLA	CNN	1215.4	22.22	87.1%
	Att- Bi-LSTM	1299.6	24.42	83.8%
	R-Bert	1486.7	28.92	81.4%
(TFIDF+QB)-CLA	CNN	376	6.54	91.5%
	Att- Bi-LSTM	386	6.77	93.6%
	R-Bert	395	11.52	92.7%
(TFIDF+QB)-WLA	CNN	376	6.66	94.3%
	Att- Bi-LSTM	386	7.21	91.2%
	R-Bert	395	7.67	90.8%

The fifth column shows the attack success percentage for 500 sentences in SemEval-2010 Task 8 (Table 4), and 1000 sentences in TACRED (Table 5) against the corresponding target NRE models. This indicates the number of sentences for which the attack model was successful in generating adversarial sentences. It is noticed that, on the SemEval-2010 Task 8 dataset, the combined attack TFIDF-QB generated more successful adversarial sentences, that is, 91.5%, 93.6%, and 92.7%, for CLA, and 94.3%, 91.2%, and 90.8% for WLA (Table 4). The corresponding results for the TACRED dataset were 97.4%, 96.2%, 95.5% (CLA), and 95.3%, 92.4%, and 92.1% (WLA), as seen in Table 5. This attack success rate is significantly higher than those of the TFIDF and QB methods. The low success rate of TFIDF is caused by missing important words in the TFIDF-list, implying that this method may be unable to determine the important words in all test sentences. To overcome this, the combined method TFIDF-QB uses the QB method to obtain the remaining important words. Accordingly, invoking the classifier only a few hundred times using TFIDF-QB can provide better and more successful attacks in supervised RE. The column “AvgTime” in both tables shows the time required for generating one adversarial sentence. Here, it can also be noticed that TFIDF (CLA and WLA) required significantly less time to complete both tasks (2–4 and 5–6 s) than the other two methods QB (CLA, WLA) (24–28 and 43–48 s) and TFIDF-QB+(CLA, WLA) (6–11 and 11–17 s). The average sentence length in TACRED is almost twice as much as that of SemEval-2010 Task 8, and thus the average time required to generate a single adversarial sentence is also almost twice as much. All these columns are correlated. For example, if an attack rarely invokes the classifier to determine significant words, the time to generate one successful adversarial sentence is shorter. TFIDF is effective in generating adversarial sentences in a short time, but the attack success is rather low. The QB-based method can determine almost all significant words, but it requires more time and invokes the classifier a few thousand times. Therefore, TFIDF-QB (CLA, WLA) exploits the advantages of both the TFIDF and QB methods to achieve high attack success on NRE models in a short time.

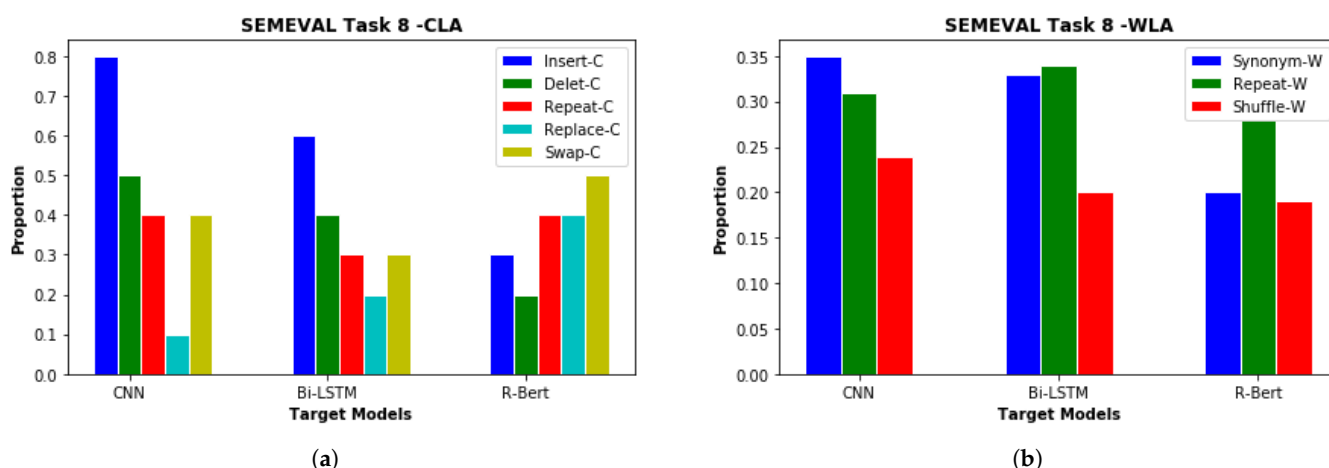
**Table 5.** Results of adversarial attack on 1000 sentences from TACRED and corresponding targeted models.

Attack Types	Victim Models	Invok# ( $WI_m$ )	AvgTime(s)	Attack Success
TFIDF-CLA	PA-LSTM	0	5.82	80.7%
	C-GCN+PA-LSTM	0	6.10	79.8%
	Span-Bert	0	5.31	77.3%
TFIDF-WLA	PA-LSTM	0	6.7	79.1%
	C-GCN+PA-LSTM	0	6.42	77.4%
	Span-Bert	0	5.55	76.3%
QB-CLA	PA-LSTM	2571.2	45.74	88.4%
	C-GCN+PA-LSTM	2719.8	47.46	86.3%
	Span-Bert	3018.9	47.97	82.2%
QB-WLA	PA-LSTM	2571.2	43.12	85.2%
	C-GCN+PA-LSTM	2719.8	45.23	83.6%
	Span-Bert	3018.9	48.85	82.7%
(TFIDF+QB)-CLA	PA-LSTM	792	11.51	97.4%
	C-GCN+PA-LSTM	778	13.72	96.2%
	Span-Bert	802	15.54	95.5%
(TFIDF+QB)-WLA	PA-LSTM	792	13.91	95.3%
	C-GCN+PA-LSTM	778	15.23	92.4%
	Span-Bert	802	16.82	92.1%

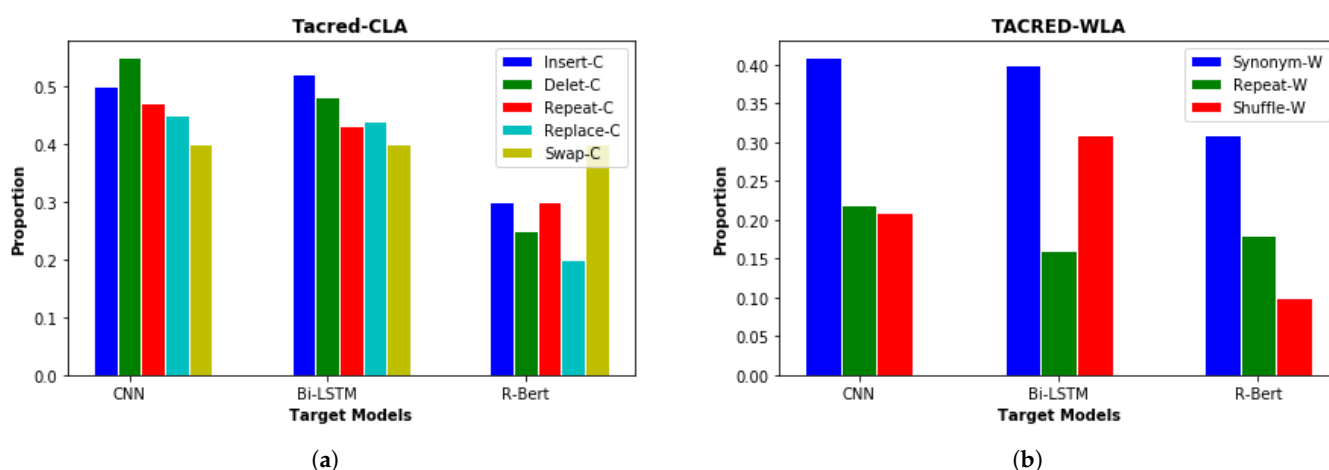


### 6.1. Comparison of Perturbation Types

The distributions of all types of generated perturbations for both datasets are shown in Figures 2 and 3. Figure 2a,b show the distribution of CLAs and WLAs for SemEval-2010 Task 8. It can be seen that Insert-C was more effective for CLA-type perturbations in CNN and attention BI-LSTM, whereas Swap-C performed better on R-Bert; however, overall, the attack success of all methods on BERT remained lower than on the other models. In WLA, Synonym-W affected the CNN and attention BI-LSTM almost as the Repeat-W operation; however, for R-Bert, Repeat-W proved to be a more effective attack.



**Figure 2.** (a) shows the distribution of Character Level Attacks (CLA's) perturbations and (b) shows the distribution of World Level Attacks (WLA's) perturbation on 500 sentences of SemEval-2010 Task 8 data set.



**Figure 3.** (a) shows the distribution of Character Level Attacks (CLA's) perturbations and (b) shows the distribution of World Level Attacks (WLA's) perturbation on 1000 sentences of TACRED data set.

Similarly, Figure 3c,d show the distributions of both types of perturbation (CLA and WLA) for the TACRED dataset. The classifiers of the TACRED datasets were easily fooled by almost every type of character-level perturbation. Synonym-W in WLA proved to be the most effective in this case, indicating that the training set of TACRED did not repeatedly use synonyms of important words in different sentences of the same class, whereas in SemEval-2010 Task 8, words such as “caused” and its synonyms such as “resulted” have been used on different occasions. This explains the reduced effectiveness of Synonym-W on SemEval-2010 Task 8.

Figures 2b and 3d show the WLA distribution on the target models for both datasets. Figure 2b shows that the Synonym-W and Repeat-W (repeat non-significant words) operations are dominant in attacks on CNN and Bi-LSTM for SemEval-2010 Task 8. However, their effect is not significantly stronger because the sentences in SemEval-2010 Task 8 are shorter and less complicated than those in TACRED. The Swap-W operation was the least dominant. In the case of the TACRED dataset, all three models were fooled by synonym replacement. Here, it should be noticed that TACRED is quite larger than SemEval-2010 Task 8. Moreover, not all synonyms are used, but only a few important words for classification are repeatedly used in each type of sentence.

## 6.2. Comparison with Modified Baseline Black-Box Attacks of Simple Text Classification Tasks

As mentioned earlier, our work is the first to evaluate the sustainability of supervised RE DNNs. Black-box adversarial attacks designed for other text classification problems cannot be directly applied to RE datasets to generate adversarial sentences. In RE datasets, each sentence has mentioned entities ‘e1’ and ‘e2’. The relation is to be extracted between these entities, which the attacker is not supposed to alter during the generation of an adversarial sentence, by doing so, the attack RE will lose its meaning. In Algorithm 1 Line (3), we used conditions for not making any changes to the mentioned entities of the sentence. In this case, if we want to apply the baseline black-box attacks of other text classification tasks to RE, then those algorithms need a slight modification in the same way, so that the mentioned entities are not altered during the generation of adversarial sentences.

The baselines that we modified for comparison to our RE problems are described below:

1. **PSO:** It uses substitution based on sememe and particle swarm optimization. It is a score-based attack [54].
2. **Textfooler:** It ranks the words using the confidence score of targeted victim model and replaced those words with synonyms [50].
3. **PWWS:** This approach used the confidence score of models and rank them accordingly. It uses WordNet for substituting the words [55].

Table 6 shows the comparison of modified baselines with our specialised TFIDF+QB-WLA attack for RE. Our model fully outperformed other attack models in terms of query efficiency (column:Invoke#(WIm)) by generating adversarial sentence in only a few queries to the models, therefore saving processing time. In terms of after attack accuracy (column:acc%) our model brought down the accuracy of PA-LSTM, C-GCN+PA-LSTM, CNN and Att Bi-LSTM better than PSO, TF and PWWS while for R-BERT and Span-BERT TF performed better. The column pert% shows the number of words perturbed for generating adversarial sentences. our model shows lower percentage of words perturbed while attacking PA-LSTM and Att-Bi-LSTM.

## 6.3. Adversarial Sentence Examples

Table 7 shows two successful adversarial sentences for SemEval-2010 Task 8, and two for TACRED. The first changed the prediction of the classifier from 92.3% “cause-effect” to 73.4% “other” by only using two character-level perturbations. “**Resulted**” was changed to “**res ulted**”, but a space between “s” and “u” was inserted using the Insert-C operation, and “**bombs**” was changed to “**b0mbs**” by the Replace-C operation. The second adversarial sentence changed the prediction from 87.4% “entity-origin” to 82.1% “product-producer” by using two types of WLA: Synonym-W replaced the word “constructed” with “manufactured”, and “from” with “against”.

Similarly, in TACRED, the first adversarial sentence changed the prediction of the original label “Per:Spouse” 92.4% to “Per:other\_Family” 84.2 % by the WLA operations Synonym-W and Swap-W. “**wife**” and “**husband**” were replaced with “**bride**” and “**hubby**,” respectively, and “**appeared**” was swapped with the “**hubby**.” The second adversarial sentence for TACRED was generated by three CLA perturbations, that is, Replace-C, Swap-C, and Insert-C, by changing “**Attended**” to “**atteNded**”, “**received**” to “**recieved**”, and

*“professor”* to *“proffessor”*. This perturbation reduced the accuracy of the original label “Per:School\_attended” 88.9% to “Per:Other\_Family” 77.9%.

#### 6.4. Human Evaluation

We performed a human evaluation test to determine whether the adversarial sentences generated for the RE task were easy to recognize. Ten graduate-level students were selected as judges to score the reading fluency and verify whether the generated adversarial examples were semantically similar to the originals. We selected 100 adversarial sentences generated in the R-Bert model for SemEval-2010 Task 8, and 100 adversarial sentences generated in SpanBert for the TACRED dataset. Subsequently, we mixed them with their corresponding 100 original sentences for each dataset. The judges were divided into two groups of five individuals each. We asked the judges to score the similarity between the original and adversarial sentences on a [1–5] Likert scale indicating the likelihood of adversarial sentences being modified by a machine. Table 8 shows a comparison of automatic evaluation with human evaluation. It can be seen that the models are misclassified at a high rate, but the human classification is almost similar to that of the original classification. The Likert-scale score is slightly higher for the machines that generate adversarial sentences than for those generating the original sentences because the judges think that there are minor changes in synonyms or spellings in the original words. Nevertheless, the machine evaluation of the original text and human evaluation of adversarial texts are quite close, implying that it remains challenging for humans to perceive the modifications.

**Table 6.** Comparison of modified word-level black-box adversarial attacks of other text classification tasks on 1000 and 500 sentences of TACRED and SemEval-2010 Task 8 datasets with our combined TFIDF+QB-WLA.

Dataset	Attack	PA-LSTM				C-GCN+PA-LSTM				Span-BERT			
		Orig%	Acc.%	Pert.%	Invoke # W(Im)	Orig%	Acc.%	Pert.%	Invoke # W(Im)	Orig%	Acc.%	Pert.%	Invoke # W(Im)
Tacred	PSO	62.6%	14.9	15.8	2588	64.5	15.7	14.8	2774	69.1	18.2	18.9	2917
	TF		15.3	16.1	2577		15.4	14.4	2766		19.7	18.1	2905
	PWWS		15.7	17.2	2601		16.1	15.2	2892		22.8	20.4	3024
	Ours		14.1	13.7	792		13.1	14.9	778		18.4	18.2	802
Dataset	Attack	CNN				Att-Bi-LSTM				R-BERT			
		Orig%	Acc.%	Pert.%	Invoke # W(Im)	Orig%	Acc.%	Pert.%	Invoke # W(Im)	Orig%	Acc.%	Pert.%	Invoke # W(Im)
SemEval 2010 Task 8	PSO	81.2	15.7	8.9	1201	83.4	17.7	11.4	1229	87.7	21.1	12.4	1349
	TF		16.6	10.2	1193		18.7	11.7	1211		19.2	13.2	1376
	PWWS		19.9	13.4	1225		21.8	14.1	1272		26.2	14.5	1462
	Ours		15.2	9.3	376		17.4	11.1	386		20.6	14.2	395

**Table 7.** Adversarial examples for SemEval-2010 Task 8 and TACRED. Changing a fraction of the words in a sentence with adversarially generated bugs (WLA and CLA) misleads the classifier to yield incorrect outputs. The new sentence preserves most of the original meaning and is correctly classified by humans although it contains small perturbations.

SemEval-2010 Task 8, Adversarial Sentence Examples	
<b>Original sentence</b> <b>Relation: cause–effect, 92.3%</b>	The <e1>airstrike</e1>also resulted in several secondary<e2>explosions</e2>, leading Marines at the site to suspect that the house may have contained
Adversarial sentence CLA Relation: other, 73.4%	The <e1>airstrike</e1>also <i>resulted</i> resulted in several secondary <e2>explosions</e2>, leading Marines at the site to suspect that the house may have contained homemade <i>bombs</i> bombs.
<b>Original sentence</b> <b>Relation: entity–origin, 87.4%</b>	This <e1>paper</e1>is constructed from a portion of a <e2>thesis</e2>presented by Edward W. Shand, June, 1930, for the degree of Doctor of Philosophy at New York University.”
Adversarial sentence WLA Relation: product–producer, 82.1%	This <e1>paper</e1>is manufactured <i>constructed</i> against <i>from</i> a portion of a <e2>thesis</e2>presented by Edward W. Shand, June, 1930, for the degree of Doctor of Philosophy at New York University.”
TACRED, adversarial sentence examples	
<b>Original sentence</b> <b>Relation: per:spouse 92.4%</b>	In a second statement read to the inquest jury, Jupp’s wife Pat said her husband appeared to have realized instantly his injuries would likely be fatal—asking a colleague to call her and tell her he loved her. (Subj: Jupp, Obj: Pat)
Adversarial Sentence- WLA Relation: Per: other_Family, 84.2%	In a second statement read to the inquest jury, Jupp’s <i>wife</i> bride Pat said her <i>husband</i> appeared hubby <i>appeared</i> to have realized instantly his injuries would likely be fatal—asking a colleague to call her and tell her he loved her. (Subj: Jupp, Obj: Pat)
<b>Original Sentence</b> <b>Relation: Per: school_attended, 88.9%</b>	He attended Princeton University and then the University of California, where he received a Ph.D. in 1987 and was promptly hired as a professor. (Subj: He, Obj: University of California)
Adversarial sentence CLA Relation: Per: other_Family, 77.9%	He <i>attended</i> attended Princeton University and then the University of California, where he <i>received</i> recieved a Ph.D. in 1987 and was promptly hired as a <i>professor</i> professor. (Subj: He, Obj: University of California)

**Table 8.** Comparison of machine and human evaluation. Columns 4 and 5 represent the classification accuracy by the model and human, respectively. The last column represents a human evaluation of the degree to which the sentence was likely to be perturbed by a machine. A larger score indicates a higher probability.

Dataset	Model	Examples	Model Accuracy	Human Accuracy	Score [1–5]
SemEval-2010 Task 8	CNN	Original	98.1%	94.5%	1.57
		Avg adversarial	14.9%	92.3%	2.12
	Bi-LSTM	Original	88.6%	90.1%	1.80
		Avg adversarial	17.25%	88.4%	2.0
	R-Bert	Original	95.2%	98.1%	1.71
		Avg adversarial	21.55%	75.3%	2.05
TACRED	PA-LSTM	Original	82.1%	90.6%	1.42
		Avg adversarial	13.5%	89.2%	2.09
	C-GCN+PA-LSTM	Original	91.3%	95.3%	1.89
		Avg adversarial	15.9%	90.1%	2.22
	Span-Bert	Original	96.6%	94.6%	1.96
		Avg adversarial	20.7%	85.1%	2.84

## 7. Transferability

This property is popular in attacks on image classifiers. For example, adversarial images generated in a model are tested on another model to determine whether the other image classifier can be fooled as well. In text classification, it is also important to test this property, and the transferability of adversarial text between models has been studied [56,57]. However, the transferability of adversarial attacks on NRE models has not been considered. We evaluated this property by generating adversarial texts on both the RE datasets used in this study and the corresponding NRE models.

Table 9 shows the results of the experiment. It can be seen that transferability is quite moderate. For SemEval-2010 Task 8, the attack success rates are not as high as on TACRED, as the classifiers of SemEval-2010 Task 8 are smarter and more robust than those of TACRED. This is because TACRED is a new dataset, and the classifier has only obtained a few high results for this dataset compared with SemEval-2010 Task 8. The transferability of BERT-based models is quite high. For example, a value of 81.4% is achieved in the case of the PA-LSTM model on adversarial sentences generated for TACRED. This evaluation demonstrates that the adversarial sentences generated by the proposed method are highly transferable across all the other models.

**Table 9.** Transferability of generated adversarial sentences between targeted models for SemEval-2010 Task 8 and TACRED.

		CNN	Att Bi-LSTM	R-BERT
SemEval-2010 Task 8	CNN	97.4%	66.7%	32.9%
	Att Bi-LSTM	71.8%	96.4%	31.5%
	R-BERT	78.1%	69.4%	94.5%
		PA-LSTM	C-GCN+PA-LSTM	SpanBERT
TACRED	PA-LSTM	99.2%	76.9%	52.3%
	C-GCN+PA-LSTM	91.4%	97.2%	58.6
	SpanBert	81.4%	79.5%	90.2%



## 8. Defense Strategies

Herein, we discuss potential defense strategies that can be applied to improve the robustness of these deep neural network models. There are techniques such as spell checkers and adversarial training. In [15,33], it was demonstrated that deep neural models remain vulnerable to misspelled words even after a spell checker is applied. We applied two methods to improve the robustness of the targeted models: spell checking and adversarial training.

### 8.1. Spell Checker

We applied the Google spell checker to the character-level perturbations, and we analyzed which misspelled words or noise were easily detected by the spell checker. The pie chart shows the correction rate for misspelled or altered words. It can be seen that the Insert-C (37.0%) and Delete-C (30%) error types are the easiest to detect, whereas Replace-C errors are not easily detected because this operation replaces a word by visually similar characters. For example, if we replace “a” by “@” between the first and last characters, the spell checker cannot detect the replacement. The spell checker is not suitable for adversarial sentences generated by WLAs, as words are not misspelled. The attack success results are given in Table 10. It can be seen that the spell checker reduces the attack success rate, but the attack success rate on TACRED is higher than on SemEval-2010 Task 8, because the accuracy of the best model on the former dataset is approximately 71%. This implies that the classification models require substantial improvement. BERT proved to be more robust than CNN, LSTM, and GCN-based models.

**Table 10.** Success rate of adversarial attack after spell checking in the case of character-level perturbation.

	Attack Success Rate		
	CNN	Att Bi-LSTM	R-Bert
<b>SemEval-2010 Task 8</b>	26.4%	24.3%	18.9%
<b>TACRED</b>	PA-LSTM	C-GCN+PA-LSTM	SpanBert
	39.4%	38.2%	32.7%

### 8.2. Adversarial Training

Adversarial training is another defense method that has long been used in image-based adversarial attacks. Moreover, this method has been adopted to defend classifiers against adversarial attacks in various text classification tasks. In adversarial training, models become more robust by adding adversarial sentences generated by adversarial attacks to the training set. In our experiment, we selected 5000 and 2000 sentences from the training sets of TACRED and SemEval-2010 Task 8, respectively, and generated adversarial sentences with both character- and word-level perturbations. Subsequently, we trained each corresponding model using these mixed datasets. The performance of each model before and after adversarial training on the test sets is shown in Table 11, which also shows the average number of perturbed words per sentence for CLA and WLA before and after adversarial training. It can be noted that after adversarial training, model accuracy on the adversarial sentences generated from the test sets increased. Moreover, the BERT-based model became more robust through adversarial training than all other models, and the average percentage of perturbed words also increased, demonstrating it was difficult to fool the models using originally perturbed words. Accordingly, it can be concluded that adversarial training can improve the accuracy of targeted models to some degree against adversarial attacks. However, adversarial training has the limitations that the attack strategies are unknown, and the number of adversarial sentences for training is limited. This is particularly relevant in practice, as attackers do not make their strategies and adversarial texts public.

**Table 11.** Success rate of adversarial attack after adversarial training.

Dataset	SemEval-2010 Task 8				TACRED	
Model	CNN	Att- Bi-LSTM	R-Bert	PA-LSTM	C-GCN+PA-LSTM	SpanBert
Original TFIDF+QB-CLA	14.6%	17.1%	22.5%	12.9%	13.1%	18.4%
+Adv-training	29.4%	26.1%	<b>32.6%</b>	27.3%	32.7%	<b>35.3%</b>
(Original) TFIDF+QB-WLA	15.2%	17.4%	20.6%	14.1%	18.7%	23.1%
+Adv-training	33.3%	35.2%	<b>38.8%</b>	27.6%	32.3%	<b>36.5%</b>
Perturbed words (Avg)	9.3%	11.1%	14.2%	13.7%	15.1%	18.2%
Af. Perturbed words (Avg)	12.3%	15.1%	17.2%	17.6%	19.3%	22.1%

## 9. Conclusions and Future Work

We studied adversarial attacks against two popular RE datasets to evaluate the robustness of six representative deep learning NRE models under black-box settings and prove its feasibility and sustainability. It was experimentally demonstrated that no open source NRE model is sustainable and robust against character and word-level adversarial attacks. The proposed TFIDF method is efficient, fast, and effective in generating adversarial sentences, and that the combined (TFIDF-QB)-based method reduced attack time by minimizing the number of queries. Human evaluation demonstrated that the adversarial sentences were legible and imperceptible. Furthermore, the proposed defense strategies (spell checker and adversarial training) have the possibility of improving model robustness. We believe that our findings will aid in the development of more robust RE classifiers.

In the future, we aim to find new attack methods under black-box settings to evaluate document level RE tasks and also enhance the sustainability of other NLP related tasks.

**Author Contributions:** Conceptualization, I.U.H. and Z.Y.K.; methodology, I.U.H. and Z.Y.K.; software, I.U.H.; validation, I.U.H., A.A., and A.K.; formal analysis, A.A., B.H., Y.-E.L., and K.-I.K.; investigation, I.U.H.; resources, I.U.H., A.A., and K.-I.K.; data curation, I.U.H., Z.Y.K., and A.A.; writing—original draft preparation, I.U.H. and A.K.; writing—review and editing, A.A., Y.-E.L., and B.H.; visualization, I.U.H.; supervision, K.-I.K. and A.A.; project administration, K.-I.K. and A.A.; funding acquisition, K.-I.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by an Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01343, Training Key Talents in Industrial Convergence Security).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data available on request.

**Conflicts of Interest:** The author declared no conflict of interest.

## References

- Li, Q.; Li, L.; Wang, W.; Li, Q.; Zhong, J. A comprehensive exploration of semantic relation extraction via pre-trained CNNs. *Knowl.-Based Syst.* **2020**, *194*, 105488. [\[CrossRef\]](#)
- Yao, X.; Van Durme, B. Information extraction over structured data: Question answering with freebase. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, MD, USA, 22–27 June 2014; pp. 956–966.
- Wu, F.; Weld, D.S. Open information extraction using Wikipedia. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010; pp. 118–127.
- Khan, Z.; Niu, Z.; Yousif, A. Joint Deep Recommendation Model Exploiting Reviews and Metadata Information. *Neurocomputing* **2020**, *402*, 256–265. [\[CrossRef\]](#)
- Khan, Z.Y.; Niu, Z.; Nyamawe, A.S.; Haq, I. A Deep Hybrid Model for Recommendation by jointly leveraging ratings, reviews and metadata information. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104066. [\[CrossRef\]](#)

6. Hendrickx, I.; Kim, S.N.; Kozareva, Z.; Nakov, P.; Séaghdha, D.O.; Padó, S.; Pennacchiotti, M.; Romano, L.; Szpakowicz, S. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Stroudsburg, PA, USA, 15–16 July 2010; pp. 33–38.
7. Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; Manning, C.D. Position-aware attention and supervised data improve slot filling. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 35–45.
8. Wang, H.; Qin, K.; Lu, G.; Luo, G.; Liu, G. Direction-sensitive relation extraction using Bi-SDP attention model. *Knowl.-Based Syst.* **2020**, *198*, 105928. [\[CrossRef\]](#)
9. Khan, Z.Y.; Niu, Z.; Sandiwarno, S.; Prince, R. Deep learning techniques for rating prediction: A survey of the state-of-the-art. *Artif. Intell. Rev.* **2020**, *54*, 1–41. [\[CrossRef\]](#)
10. Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; Mittal, A. Evaluating adversarial attacks against multiple fact verification systems. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 2937–2946.
11. Jia, R.; Liang, P. Adversarial Examples for Evaluating Reading Comprehension Systems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 2021–2031. [\[CrossRef\]](#)
12. Poliak, A.; Naradowsky, J.; Haldar, A.; Rudinger, R.; Van Durme, B. Hypothesis Only Baselines in Natural Language Inference. In Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, New Orleans, LA, USA, 5–6 June 2018; Association for Computational Linguistics: New Orleans, LA, USA, 2018; pp. 180–191. [\[CrossRef\]](#)
13. Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S.; Smith, N.A. Annotation Artifacts in Natural Language Inference Data. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, LA, USA, 1–6 June 2018; Association for Computational Linguistics: New Orleans, LA, USA, 2018; pp. 107–112. [\[CrossRef\]](#)
14. Mudrakarta, P.K.; Taly, A.; Sundararajan, M.; Dhamdhare, K. Did the Model Understand the Question? In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 1896–1906. [\[CrossRef\]](#)
15. Li, J.; Tao, C.; Peng, N.; Wu, W.; Zhao, D.; Yan, R. Evaluating and Enhancing the Robustness of Retrieval-Based Dialogue Systems with Adversarial Examples. In *CCF International Conference on Natural Language Processing and Chinese Computing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 142–154.
16. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2014**, arXiv:1412.6572.
17. Carlini, N.; Wagner, D. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; pp. 1–7.
18. Li, J.; Monroe, W.; Jurafsky, D. Understanding Neural Networks through Representation Erasure. *arXiv* **2016**, arXiv:1612.08220.
19. Bhagoji, A.N.; He, W.; Li, B.; Song, D. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 158–174.
20. Ilyas, A.; Engstrom, L.; Athalye, A.; Lin, J. Black-box Adversarial Attacks with Limited Queries and Information. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Dy, J., Krause, A., Eds.; PMLR, Stockholmsmässan: Stockholm, Sweden, 2018; Volume 80, pp. 2137–2146.
21. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation Classification via Convolutional Deep Neural Network. In Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; Dublin City University and Association for Computational Linguistics: Dublin, Ireland, 2014; pp. 2335–2344.
22. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Melbourne, Australia, 15–20 July 2016; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 207–212. [\[CrossRef\]](#)
23. Wu, S.; He, Y. Enriching pre-trained language model with entity information for relation classification. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 September 2019; pp. 2361–2364.
24. Zhang, Y.; Qi, P.; Manning, C.D. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 2205–2215. [\[CrossRef\]](#)
25. Joshi, M.; Chen, D.; Liu, Y.; Weld, D.S.; Zettlemoyer, L.; Levy, O. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 64–77. [\[CrossRef\]](#)
26. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.J.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2014**, arXiv:1312.6199.
27. 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, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden, 13–19 July 2018; pp. 4208–4215. [\[CrossRef\]](#)

28. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582.
29. Nguyen, A.; Yosinski, J.; Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 427–436.
30. 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; pp. 506–519.
31. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, USA, 7–12 December 2015; pp. 649–657.
32. Pasi, G.; Piwowarski, B.; Azzopardi, L.; Hanbury, A. (Eds.) Lecture Notes in Computer Science. In Proceedings of the Advances in Information Retrieval—40th European Conference on IR Research, ECIR 2018, Grenoble, France, 26–29 March 2018; Springer: Berlin/Heidelberg, Germany, 2018; Volume 10772. [[CrossRef](#)]
33. 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 (Volume 2: Short Papers), Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 31–36. [[CrossRef](#)]
34. Belinkov, Y.; Bisk, Y. Synthetic and Natural Noise Both Break Neural Machine Translation. *arXiv* **2017**, arXiv:1711.02173.
35. Ebrahimi, J.; Lowd, D.; Dou, D. On Adversarial Examples for Character-Level Neural Machine Translation. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–25 August 2018; Association for Computational Linguistics: Santa Fe, NM, USA, 2018; pp. 653–663.
36. Li, Y.; Cohn, T.; Baldwin, T. Robust Training under Linguistic Adversity. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Valencia, Spain, 3–7 April 2017; Association for Computational Linguistics: Valencia, Spain, 2017; pp. 21–27.
37. Xie, Z.; Wang, S.I.; Li, J.; Lévy, D.; Nie, A.; Jurafsky, D.; Ng, A.Y. Data Noising as Smoothing in Neural Network Language Models. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
38. Iyyer, M.; Manjunatha, V.; Boyd-Graber, J.; Daumé III, H. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; Association for Computational Linguistics: Beijing, China, 2015; pp. 1681–1691. [[CrossRef](#)]
39. Mahler, T.; Cheung, W.; Elsner, M.; King, D.; de Marneffe, M.C.; Shain, C.; Stevens-Guille, S.; White, M. Breaking NLP: Using Morphosyntax, Semantics, Pragmatics and World Knowledge to Fool Sentiment Analysis Systems. In Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems, Copenhagen, Denmark, 8 September 2017; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 33–39. [[CrossRef](#)]
40. Staliūnaitė, I.; Bonfil, B. Breaking sentiment analysis of movie reviews. In Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems, Copenhagen, Denmark, 8 September 2017; pp. 61–64.
41. Burlot, F.; Yvon, F. Evaluating the morphological competence of Machine Translation Systems. In Proceedings of the Second Conference on Machine Translation, Copenhagen, Denmark, 7–8 September 2017; pp. 43–55.
42. Isabelle, P.; Cherry, C.; Foster, G. A Challenge Set Approach to Evaluating Machine Translation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 2486–2496. [[CrossRef](#)]
43. Levesque, H.J. On Our Best Behaviour. *Artif. Intell.* **2014**, *212*, 27–35. [[CrossRef](#)]
44. Naik, A.; Ravichander, A.; Sadeh, N.; Rose, C.; Neubig, G. Stress Test Evaluation for Natural Language Inference. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–25 August 2018; Association for Computational Linguistics: Santa Fe, NM, USA, 2018; pp. 2340–2353.
45. Xiang, T.; Liu, H.; Guo, S.; Zhang, T.; Liao, X. Local Black-box Adversarial Attacks: A Query Efficient Approach. *arXiv* **2021**, arXiv:2101.01032.
46. Ilyas, A.; Engstrom, L.; Athalye, A.; Lin, J. Query-Efficient Black-box Adversarial Examples. *arXiv* **2017**, arXiv:1712.07113.
47. Cheng, M.; Singh, S.; Chen, P.H.; Chen, P.Y.; Liu, S.; Hsieh, C.J. Sign-OPT: A Query-Efficient Hard-label Adversarial Attack. *arXiv* **2019**, arXiv:1909.10773.
48. Shen, Y.; Huang, X. Attention-Based Convolutional Neural Network for Semantic Relation Extraction. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; The COLING 2016 Organizing Committee: Osaka, Japan, 2016; pp. 2526–2536.
49. Li, J.; Ji, S.; Du, T.; Li, B.; Wang, T. TextBugger: Generating Adversarial Text Against Real-world Applications. In Proceedings of the 26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, CA, USA, 24–27 February 2019.
50. 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 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.

51. Mrkšić, N.; Séaghdha, D.; Thomson, B.; Gašić, M.; Rojas-Barahona, L.; Su, P.H.; Vandyke, D.; Wen, T.H.; Young, S. Counter-fitting Word Vectors to Linguistic Constraints. *arXiv* **2016**, arXiv:1603.00892.
52. Hill, F.; Reichart, R.; Korhonen, A. SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Comput. Linguist.* **2015**, *41*, 665–695. [[CrossRef](#)]
53. Cer, D.; Yang, Y.; Kong, S.Y.; Hua, N.; Limtiaco, N.; St. John, R.; 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; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 169–174. [[CrossRef](#)]
54. 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, ACL 2020, Online, 5–10 July 2020; Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R., Eds.; Association for Computational Linguistics: Online, 2020; pp. 6066–6080. [[CrossRef](#)]
55. 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; Association for Computational Linguistics: Florence, Italy, 2019; pp. 1085–1097. [[CrossRef](#)]
56. 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.* **2020**, *11*, 1–41. [[CrossRef](#)]
57. Alshemali, B.; Kalita, J. Improving the reliability of deep neural networks in NLP: A review. *Knowl.-Based Syst.* **2020**, *191*, 105210. [[CrossRef](#)]