Computer Systems Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.040159

ARTICLE

An Efficient Character-Level Adversarial Attack Inspired by Textual Variations in Online Social Media Platforms

Jebran Khan[1], Kashif Ahmad[2] and Kyung-Ah Sohn[1,3,*]

[1]Department of Artificial Intelligence, Ajou University, Suwon, Korea

[2]Department of Computer Science, Munster Technological University, Cork, Ireland

[3]Department of Software and Computer Engineering, Ajou University, Suwon, Korea

*Corresponding Author: Kyung-Ah Sohn. Email: kasohn@ajou.ac.kr

Received: 07 March 2023 Accepted: 17 May 2023 Published: 09 November 2023

ABSTRACT

In recent years, the growing popularity of social media platforms has led to several interesting natural language processing (NLP) applications. However, these social media-based NLP applications are subject to different types of adversarial attacks due to the vulnerabilities of machine learning (ML) and NLP techniques. This work presents a new low-level adversarial attack recipe inspired by textual variations in online social media communication. These variations are generated to convey the message using out-of-vocabulary words based on visual and phonetic similarities of characters and words in the shortest possible form. The intuition of the proposed scheme is to generate adversarial examples influenced by human cognition in text generation on social media platforms while preserving human robustness in text understanding with the fewest possible perturbations. The intentional textual variations introduced by users in online communication motivate us to replicate such trends in attacking text to see the effects of such widely used textual variations on the deep learning classifiers. In this work, the four most commonly used textual variations are chosen to generate adversarial examples. Moreover, this article introduced a word importance ranking-based beam search algorithm as a searching method for the best possible perturbation selection. The effectiveness of the proposed adversarial attacks has been demonstrated on four benchmark datasets in an extensive experimental setup.

KEYWORDS

Adversarial attack; text classification; social media; character-level attack; phonetic similarity; visual similarity; word importance rank; beam search

1 Introduction

In the modern world, social media outlets, such as Twitter, Facebook, and Instagram, have become a major source of communication and information dissemination [1]. A wide audience and global accessibility are the key attributes that make such platforms a preferred choice for a diversified set of application domains. Thanks to natural language processing (NLP), computer vision (CV), and machine learning (ML) algorithms, meaningful information can be extracted for social media data. However, social media-based NLP applications are subject to various adversarial attacks. The key

factors that influence the growing interest in attacking these applications include the dependency of these applications on automated decisions and the level of impact such attacks can have [2].

Adversarial attacks are designed to fool ML models while preserving human robustness. To this aim, attackers yield adversarial data samples by modifying instances of actual data samples from a dataset. Constructing an adversarial sample generally involves replacing 10%–30% of words in a sentence with synonyms or out-of-vocabulary (OOV) words that do not change their meaning. These adversarial examples are generated such that the model's output over the perturbed input is incorrect with a high confidence score.

The typical characteristic of a quality adversarial attack is to cause maximal damage to the machine's performance with minimal effect on the human understanding of the generated text [3–5]. In NLP, unlike CV, where pixel-level attacks may lead to catastrophic failure, producing a suitable and efficient adversarial attack is more challenging. In NLP, some existing word-level attacks are designed to delete words [6], substitute words by their synonyms and hyponyms [7], and paraphrase sentences [8]. However, these high-level attacks are not suitable in real-world social media based-NLP applications, such as spam generation, short-text generation, social media communication, and online conversations, due to the user's ignorance of the training data, lack of vocabulary, and the classification model architecture. In contrast, low-level adversarial attacks result in OOV words by simple character-level perturbation, which is harder to detect and recognize. These perturbations include random character insertion, deletion, substitution, and transposition [9–13]. Although significant, these are not used in practice due to their limitations in generating simple editing modifications, effectiveness, replication of human cognition of text generation, and naturalness.

This work proposes a framework that effectively generates utility-preserving adversarial samples against state-of-the-art classification models under untargeted black-box settings. This article presents a set of perturbations inspired by text generation trends on online social networks, where intended messages are usually conveyed in a short text. Thus, to generate adversarial samples, social media text is perturbed by following the techniques used for minimizing the text. However, during the process, it is made sure that the perturbed text is either visually or phonetically similar to actual text so that the message is received without losing human robustness while fooling the filters. Moreover, a word importance ranking-based beam search mechanism is introduced to select the most suitable perturbations. This method first ranks the elements in the input string based on their importance and then chooses a beam of b words for perturbation. The process is repeated until the best combination of perturbations is obtained.

The main contributions of this study are summarized as follows:

- To provide a hybrid perturbation method that mimics the social media textual variations to successfully attack social media-based NLP applications while preserving human and utility robustness.
- Introduce a word saliency-based beam search method for an efficient and effective adversarial attack in a black-box setting.
- Evaluate the effectiveness of the proposed adversarial attack on state-of-the-art text classification models and benchmark datasets in terms of attack success rate, perturbation rate, and preservation of semantic similarity.
- Discuss the potential defense strategies against the proposed attack with preliminary results.

The rest of the paper is organized as follows. Section 2 provides a detailed overview of related work. Section 3 describes the proposed adversarial attack. Section 4 details the experimental setup and the datasets used for the evaluation. Section 5 provides the experimental results. Section 6 discusses some potential defense mechanisms, and Section 7 concludes the paper.

2 Related Study

The literature on adversarial ML is very rich. Most of the initial efforts in the domain are based on image-based solutions, and relatively less attention has been paid to adversarial text analysis. However, recently, adversarial text analysis has received increased attention due to the nature and dependency of NLP applications on automated decisions [2]. The literature already reports several attractive solutions exploring different aspects of adversarial text analysis. The generation of adversarial textual examples, which is one of the critical aspects of adversarial text analysis, is widely explored in the literature [14]. For instance, Papernot et al. [15] proposed a gradient-based perturbation method against recurrent neural network (RNN) classifiers. The authors applied the modification directly to the text iteratively until the successful generation of a misleading sequence. This method is named Fast Gradient Sign Method (FGSM) as it selects a random word from the input sample and generates gradient-based perturbations corresponding to the word vector. The perturbed word vector is then mapped into words with the least Euclidean distance in the word embedding space. On the other hand, Samanta et al. [10] used the embedding gradient to estimate the importance of the words using heuristic rules and manual synonyms and typos. Liang et al. [9] used word frequencies to select a class's most important words. This method generates adversarial sequences by inserting, deleting, and modifying these critical words. These methods are applied in white-box settings as they access the model gradient and class labels for adversarial sample generations.

Adversarial attacks can be classified as high- or low-level attacks based on transformation types. High-level adversarial attacks involve insertion, deletion, replacement, and displacement of words, whereas low-level attacks involve modification at the character level. A vast majority of the literature is based on high-level adversarial attacks. For instance, Jia et al. [16] inserted semantically correct but irrelevant paragraphs into texts to fool neural reading comprehension models. Alzantot et al. [7] proposed an optimization-based method for adversarial text generation through the replacement of words in the input sequence with their nearest neighbors in the embedding space. The authors used the GloVe embedding space and a genetic algorithm for optimal solution selection. Checklist attack [17] is based on sentence contraction and extension, and name entities substitution. Similarly, TextFooler [18] which is a word-level adversarial attack, is generated through the replacement of words by their suitable synonyms and a word importance ranking-based greedy search. This method allows the generation of adversarial samples while preserving semantic similarity and syntactic correctness. Ren et al. [19] proposed another high-level attack, weighted word saliency, which replaces the most important words in the input sample with their nearest neighbors in WordNet. Zang et al. [20] used particle swarm optimization (PSO) and HowNet-based word swapping to fool the classification model.

Although effective, these rule-based adversarial attacks can generate out-of-context and complex replacements. BERT-based adversarial examples (BAE) were proposed to generate adversarial examples using the BERT-masked language model by context-aware modifications. BAE replaces and inserts tokens into the original text by masking a piece of the text and using the BERT-MLM to generate substitutes for the masked tokens. Li et al. [21] proposed BERT-Attack, attacking BERT using BERT. The authors used BERT to find the optimal solution in a huge space of possible transformation to preserve semantic consistency and fluency. The BERT-Attack uses BERT-masked

token prediction with sub-word expansion. Li et al. [22] proposed Clare by using RoBERTa masked prediction to swap, insert, and merge tokens in the input samples. Wang et al. [23] proposed a fast gradient projection method (FGPM), a synonym replacement-based adversarial attack with higher time efficiency. Goa et al. [24] proposed an adversarially regularized auto-encoder (ARAE) that maps discrete text into a continuous space and generates the adversarial examples by adding the universal adversarial perturbations in the continuous space, selecting the natural adversarial samples. A rule-based method [25] is proposed to control the number of perturbations for the word-level adversarial attacks.

The word-level attacks performed well against the state-of-the-art deep learning (DL) models; however, they are more inclined to produce samples by inserting less frequent and complex words that are rarely used in online conversations. Such adversarial examples may fool the system but fail to convey the message efficiently. The low-level adversarial attacks suit well to real-life scenarios. There exist a variety of works that propose low-level orthographic attacks. For instance, Ebrahimi et al. [26] attacked the neural network-based text classification model in a white-box setting by flipping the characters with the most adverse effects. Ebrahimi et al. [27] also fooled machine translation systems with character-level modifications. Belinkov et al. [28] used a combination of keyboard-based character swapping to replicate synthetic keyboard typos and natural typing errors captured from different Wikipedia edit histories for perturbed text generation. Hosseini et al. [29] and Rodriquez et al. [30] attacked the toxicity system by generating multi-level adversarial samples using character-level modifications for the misspelling of the abusive words and polarity shifting by inserting the word "not" to fool the system. Eger et al. [5] utilized the visual similarity of characters for perturbation. They replaced some characters in the input samples with similar-looking characters to fool the model while preserving human robustness. Tan et al. [31] attacked words by replacing them with morphological variants, which mostly resulted in orthographic attacks (in English). Gao et al. [12], and Pruthi et al. [13] used composite transformations, including character insertion, deletion, substitution, and transposition, which replicate the most common typing mistakes, to attack text classification models. Similarly, TextBugger [11] also used a composite attack combining these four typos and synonym replacements to fool the classification models. Eger et al. [32] attacked RoBERTa using nine character-level perturbations based on visual and phonetic similarities. Bhalerao et al. [33] and Le et al. [34] proposed data-driven approaches to learn and generate adversarial samples from the human-written text on the Web. They collected data from online sources and learned text generation patterns in a specific domain.

Although the above approaches have achieved good results, there is still significant room for improvement in attack success rate, naturalness, perturbation rate, and semantic consistency. The success rate of the existing character-level attacks [11–13] against transformers-based models like BERT and RoBERTa is generally low, with a high perturbation rate. The perturbations generated by these approaches are usually basic and can only be considered as replicating typing mistakes such as a single character insertion, deletion, substitution, and transposition. Other methods, such as [32–34], to attack transformer-based models while preserving human robustness, however, have non-trivial and data-dependent substitution strategies, limiting their applicability to specific tasks. This work aims to introduce a set of perturbations learned from human text generation trends and a word saliency-based beam search mechanism for optimal adversarial example selection to enhance the attack success rate. The performance of the proposed method is evaluated and compared to the existing methods based on the evaluation metrics discussed in Section 4.4. Additionally, a basic defense system is applied to check the robustness of the proposed attack against the adversarial defense.

3 Proposed Attack

This work proposes a set of perturbations inspired by trends in text generation on online social networks. In online social networks, generally, the goal is to convey the intended message with the shortest possible text. To achieve this, considering the source and target, the message must be visually or phonetically similar. Inspired by social network trends, the perturbed text is generated by relying on four transformations to fool and generate human-understandable adversarial samples in this work. The proposed algorithm is presented as Algorithm 1. The problem is formulated in the following subsection.

Algorithm 1: Algorithm for the proposed Adversarial Attack

Input: Original text space $\mathfrak{X} = \{X_1, X_2, X_3, \ldots, X_n\}$, where $X = \{w_1, w_2, w_3, \ldots, w_m\}$, a set of labels space $\mathfrak{Y} = \{Y_1, Y_2, Y_3, \ldots, Y_n\}$, and beam length b,
Output: Adversarial text X' if found

Initialize $X' \leftarrow X$
for w_i in X **do**
  Compute I_{w_i} using Eq. (3)
end for
$R_x \leftarrow Sort(\{w_1, w_2, \ldots, w_m\})$ based on I_{w_i}
$best \leftarrow R_x$
while $best \neq \varnothing$ **do**
  $X_{cand} \leftarrow \varnothing$
  for each $X_b \in best$ **do**
    for each $i \in 1, 2, \ldots, b$ **do**
      $X_{cand} \leftarrow X_{cand} \cup Transform\,(X_b, i)$
    end for
    $\boldsymbol{score}(X_{cand}) = f_y(X) - f_y\,(X_{cand})$
  end for
  if $X_{cand} \neq \varnothing$**then**
    $X_{potential} \leftarrow argmax_{X* \in X_{cand}} score\,(X*)$
    if $F\left(X_{potential}\right) \neq F(X)$ **then**
      return $X_{potential}$ as X'
    else
    $best \leftarrow$ {top b elements of X_{cand}}
    ▷ where elements are ranked based on their *score*
  end if
  else
    End search
  end if
end while
return None

3.1 Problem Formulation

Given a set of n sentences $\mathfrak{X} = \{X_1, X_2, X_3, \ldots, X_n\}$ and their corresponding labels $\mathfrak{Y} = \{Y_1, Y_2, Y_3, \ldots, Y_n\}$, we have a pre-trained classification model $\mathcal{F}$: $\mathfrak{X} \rightarrow \mathfrak{Y}$, that maps the input text

space $\mathfrak{X}$ to its corresponding label space $\mathfrak{Y}$ with confidence score $\mathcal{F}(\mathfrak{X}|\mathfrak{Y}) = \{f_1, f_2, f_3, \ldots, f_n\}$). The confidence score is the maximum posterior probability of class Y_a given X, as shown in Eq. (1).

$$f(Y_a|X_i) = argmax_{Y_i \in \mathfrak{Y}} P(Y_i|X_i) \tag{1}$$

The aim is to generate an adversarial example X' that conforms to the requirements provided in Eqs. (2) and (3) by introducing imperceptible perturbations to X, maximizing model F (.)′s loss.

$$F(X') \neq F(X) \tag{2}$$

$$L = argmax_{X_i'} \mathcal{L}\left(Y_i, f\left(X_i'\right)\right) \tag{3}$$

Here, X_i' represents the adversarial example generated by transforming X_i, $f\left(X_i'\right)$ is the model prediction score with input sample X_i', and $\mathcal{L}(.)$ is the loss function of the model.

3.2 Word Importance Ranking

From the literature, in the text classification, it is now a well-established fact that in a given instance $X = \{w_1, w_2, w_3, \ldots, w_m\}$ of m words, the level of impact of each word on the classification may vary. It is observed that a word's influence in a classification task varies, and only a few words act as influential in class prediction through model F. This observation is aligned with the findings of [35] that BERT attends to the statistical cues of some words for prediction. Therefore, it is logical to modify the words with a significant contribution to model prediction instead of random word perturbations. Selection of the most influential words maximizes the model loss while minimizing the number of perturbations and preserving the semantic similarity for the highest possible level. The most influential words are selected based on the word saliency scores, calculated as the difference between the confidence scores (f_y) of the original sample and the sample when the word w_i is replaced by unknown (UNK token). The word saliency I_{w_i} is computed as follows in Eq. (4):

$$I_{w_i} = f_y(X) - f_y\left(X_{\setminus w_i}\right) \tag{4}$$

where $X_{\setminus w_i} = \{w_1, \ldots, w_{i-1}, [UNK], w_{i+1}, \ldots, w_m\}$ is the sentence after substituting w_i with $[UNK]$.

Moreover, the stop and short words are prohibited from perturbation by removing them from the list of candidates to preserve the fluency and grammar of the perturbed text. Short words are words that are considered too short to contribute significantly to the meaning of the text. Short words are not necessarily stop words, e.g., no, so, not, hot, etc. The length of the words to be filtered out varies depending on the user's choice. In this work, words with less than three characters are considered short words. The words are then sorted in descending order based on the word's saliency scores while leaving out the stop and short words.

3.3 Transformation

The transformation for an input X returns a set of potential perturbations X' based on the transformation rules. The transformation function is independent of the goal function and constraints and returns the set of all potential transformations.

Our approach uses composite transformation for the generation of adversarial examples. In the composite attack, a set of modifications is defined where each can modify the target words. This work uses a multi-stage perturbation method to generate hybrid samples for the target words. However, it is preferred to select the transformations that maximize the model loss with minimum possible

perturbations. Fewer perturbations ensure the visual and semantic similarity of the original and adversarial samples. The transformations used in this work are discussed as follows.

3.3.1 Vowel Removal

In English, consonants are sound producers, whereas vowels are used to raise or lower the sounds of phonemes in the language. In informal communication on social networks, there is a trend of using consonants and excluding most vowels from words, except the starting vowels. This vowel removal can efficiently generate text samples with OOV tokens, which can easily fool the models, while a human can interpret them quickly with little cognitive effort [36]. The vowel removal can generate misspelled words with high phonetic similarity to the actual words, e.g., good (g ʊ d) → gd (gd).

Vowel removal is a rule-based approach implemented with constraints such that the actual phonetic representation is modified the least. These constraints include not removing the first character and not removing vowels occurring between the same characters, which unlike the existing approaches preserve the human understanding while successfully modifying the target words, e.g., "adversarial attacks" is modified to "advrsrl attcks" instead of "dvrsrl ttcks" [32]. This transformation removes vowels from the candidate words to generate adversarial samples.

Additionally, instead of removing all vowels, a schwa removal method is utilized. In English, schwa (ə) is considered the most common vowel sound. Schwa is a concise neutral sound, and it is used as a reduced vowel in many unstressed syllables. The words are converted to their equivalent phonetic representations using IPA phonetic transcription, identified the schwa, and modified the word by deleting the vowel corresponding to schwa, e.g., "consistently (kən'sistəntli)" is modified to "cnsistntly (kən'sistəntli)" instead of "cnsstntly," where only characters corresponding to shwa "ə" are removed despite all vowels.

3.3.2 Phonetic Replacement

An ideal phonetic perturbation modifies the spelling of the target word while leaving the pronunciation the same. The phonetic replacement of words, characters, and sub-words is commonly used on online platforms. It does not only occur mistakenly but also as a form of creative language use [37]. The phonetic replacement method includes two types of perturbations: a rule-based approach for substitution with similar sound characters/digits (atmosphere → 8mosphere, you → u) and mapping words with their phonetic keys such as architect → ARKTKT using a double metaphone search algorithm [38].

3.3.3 LeetSpeak

LeetSpeak is a method of text modifications characterized by character replacement with a visually similar, non-alphabet character(s), which are referred to as homoglyphs. The most commonly used homoglyphs in LeetSpeak are numbers. This work generates adversarial samples by substituting a selected set of characters, including {a, b, e, g, l, i, o, s, t} with its similar looking homoglyphs like {4, 8, 3, 9, |, 1, 0, 5/$, 7}, respectively. For instance, the word "less," "passionate," and "stylish" are modified as "l3ss," "passi0nate," and "$tylish," respectively. LeetSpeak-based substitution can be utilized to generate very complex adversarial examples by replacing a single character with multiple characters. However, keeping the motivation of human robustness, only the simplest one-with-one character mapping method is utilized.

3.3.4 Characters Insertion

Inserting unobstructive characters into words is frequently used in social media conversations, especially in the case of abusive and toxic language [38], to fool automatic harmful content detectors. The attack may minimally affect human understanding depending on the perturbation symbols used. In online social media spaces, dots, "–," "*," and "#" are the most commonly inserted symbols, where "*" is mostly inserted in slang, such as fuck → f***, etc.

Moreover, in social media emotions are often textually expressed by character insertion which is referred to as word enlargement. The word enlargement results in the OOV words which are humanly understandable, e.g., goooooooood, baaaaaaaaddd, etc. Emotional expressions are generally found in adjectives, adverbs, or interjections. Such types of emotional expressions may fool the automatic text classification systems. In this work, random spaces, dots, "–," "#," and character repetition (based on POS) are inserted to generate candidate transformations.

3.4 Search Methods

The search method successively evaluates a model and selects the best possible perturbation out of all possible transformations based on the maximum model loss. It returns the perturbation that achieves the goal and satisfies all constraints.

This work proposes a word importance ranking-based beam search (WIR-BS). In this method, the words in a given sample X are sorted according to the word importance score estimated by Eq. (3). A set of b words in order of descending saliency is considered as the first beam of candidates for perturbation, where b is the beam width representing the number of words in each iteration to find the best combination to maximize the loss function until the goal is achieved. Each word w_i in the range $i = \{1, 2, 3, \ldots, b\}$ is substituted with the best-perturbed candidate w_i', selected from all possible substitutions to maximize the loss function. The process is then repeated to generate the next set of candidates. Although the word importance estimation introduces extra computation overhead, it generates adversarial samples with computational complexity linear to the text length. However, it reduces the time complexity of a beam search from $\left(O\left(b * W^2 * T\right)\right)$ [39] to $(O\,(b * W * T))$ because it ranks the input based on its importance score and the goal will be achieved in one forward iteration. W is the number of words, and T is the number of possible transformations.

3.5 Constraints

Constraints are rules used to determine the validity of a perturbation with respect to the input sample. These constraints include pre-transformation, overlap, grammatical, and semantic constraints. These constraints can be applied based on the type of perturbations.

This work imposes several constraints to keep the perturbation rate in the minimum possible range, including word length, stop words modification, the maximum number of words, edit distance, and modification repetition. The word length, stop words modification, and repeat modification are pre-transformation constraints performed before modifying the candidate samples. The word length constraint prevents the perturbation of words that are shorter than a user-defined length. The stop words modifications constraint disallows the modification of stop words. The repeat modification constraint disallows the modification of previously modified words. The maximum number of transformed words and edit distance are considered as overlap constraints. The overlap constraints determine the perturbation validity based on the difference between the input and modified samples, i.e., the maximum number of allowed editing operations, and the maximum number of transformed words. In this work, the values of these constraints are set as follows: $minimum\ \ word\ \ length\ =\ 3$,

Levenstein edit distance = 30, *maximum number of transformed words* = 15, *stopword modification* = *set*, and *repeat modification* = *set*.

3.6 Proposed Algorithm

The overall procedure is presented in Algorithm 1, where the inputs are the set of original samples $\mathfrak{X}$, a set of their corresponding labels $\mathfrak{Y}$, and beam length b. The set $\mathfrak{X}$ consists of n sentences X_i, and $\mathfrak{Y}$ contains n labels Y_i. The proposed algorithm yields adversarial text X' corresponding to each sample in the original text upon a successful attack. The adversarial sample X' is initialized with X. Line 1 computes the importance of words I_{w_i} for each word w_i in the given sample using Eq. (3). Line 2 sorts the words in descending order of their importance ranking into R_x, where R_x is assigned to the *best* possible solution. Starting from line 3, the set of possible candidates X_{cand} is initialized as an empty set and X_b is selected as the first b elements from *best*. For each element i in X_b, subroutine *Transform* (X_b, i) selects the best transform to populate a set of possible candidates X_{cand}. The process is repeated till the end of the sentence, and the score is calculated for each set of candidates, as shown in line 6. The score for each candidate set is calculated as the difference between the confidence scores $f_y(X)$, and $f_y(X_{cand})$. Where $f_y(X)$ is the confidence score of a sample X when assigned to class y and $f_y(X_{cand})$ is the confidence score of a sample when the words in sample X are replaced with their corresponding substitutions X_{cand}. The sample with the highest score is assigned to the potential perturbation $X_{potential}$. The $X_{potential}$ is considered as the adversarial sample X' if the label of $X_{potential}$ is not the same as X, when passed through the system $F(.)$, i.e., $F\left(X_{potential}\right) \neq F(X)$.

The subroutine *Transform* (X_b, i) takes text X_b and index i as an input to generate X'_b, a set of perturbations that satisfies the given set of constraints. The perturbations are generated by using the transformation methods discussed in Section 3.3. After generating all possible replacements, a set of suitable candidates perturbations X_{cand} is generated by collecting only those transform that satisfies the defined constraints. The best replacements are selected based on the difference between the confidence score of the original text and the perturbed text from the set of possible candidates.

4 Experimental Setup

This section discusses the target ML models, datasets, baseline algorithms, and evaluation measures used in our experiments. We selected four state-of-the-art text classification models, four benchmark datasets to attack, four baseline algorithms for comparison, and five evaluation measures. The general experimental process for generating adversarial attacks on text classification models involves the following steps: data collection, model selection, definition/selection of transformations, selection of constraints, goal function definition, and definition/selection of the search method. The TextAttack [40] framework provides a platform for defining customized attacks. In this work, we define the transformation methods and a new search method, while extracting the data and pre-trained models from Huggingface using the TextAttack[1,2] framework. The following subsections provide details of each of the components of the experimental setup.

4.1 Target Models

For our experiments, four state-of-the-art models for text classification are considered. These models include convolutional neural networks (CNN), RNNs, and transformers-based text classification algorithms. Each of the following models is pre-trained on their respective datasets, e.g., the

[1] https://github.com/QData/TextAttack
[2] https://textattack.readthedocs.io/en/latest/

BERT model trained by AGNews data set is used to evaluate the robustness of the BERT model to the adversarial attacks in multi-class (4 classes) classification.

4.1.1 WordCNN

WordCNN is one of the widely used models for text classification. It consists of an embedding layer that performs $50-$ *dimensional* word embedding on $400-$ *dimensional* input vectors, a $1D$-convolutional layer composed of 100 filters of kernel size 3, a $1D$-max-pooling layer, and two fully connected layers. In this work, we rely on the implementation provided in TextAttack Model Zoo[3].

4.1.2 Long Short-Term Memory (LSTM)

LSTM is also one of the most commonly used architectures for text analysis. This work used a pre-trained one-layer bi-directional LSTM with 150 hidden state size, and a fully connected layer. The sequence length and dropout hyperparameters were set to 128 and 0.3, respectively. The model used $200-$ *dimensional* GLoVE embedding as a base. The bi-directional LSTM consists of two LSTMs where one is fed with forward and the other is fed with reverse sequence to capture the context in both directions. The output from each LSTM is concatenated before the subsequent layer. For the implementation of the model, the open library TextAttack Model Zoo[3] is utilized.

4.1.3 BERT Base Uncased

Bidirectional Encoder Representations from Transformers (BERT) [41] is one of the state-of-the-art text analysis algorithms. In this work, we attacked the pre-trained BERT base uncased as one of the target classification models available on Huggingface for all target datasets. BERT is a SOTA model, pre-trained on a large English corpus and self-supervised. The strength of this model lies in its highly pragmatic approach and its training on large datasets like Wikipedia and BookCorpus. BERT is known for its high performance in numerous downstream NLP tasks. BERT is pre-trained with two main objectives, masked language modeling (MLM) and next sentence prediction (NSP). These two objectives enable the model to learn the internal representation of the language, which is utilized to extract salient features for downstream tasks. BERT utilizes transformer encoder architecture that tokenizes and processes each token in the context of its prior and later tokens. In this work, we used the pre-trained models provided in the Huggingface library[4].

4.1.4 RoBERTa Base

RoBERTa is a state-of-the-art model introduced by [42]. RoBERTa is pre-trained on English using masked language modeling (MLM) objective. RoBERTa is a robustly optimized replication of BERT pretraining. It is pre-trained on larger data with an increased number of iterations. This work used the pre-trained models provided in the Huggingface library[5].

4.2 Data Sets

This work evaluated the proposed adversarial attack on four popular public benchmark datasets for text classification. This section provides the detail of the datasets. A summary of these datasets is provided in Table 1.

[3] https://textattack.readthedocs.io/en/latest/3recipes/models.html
[4] https://huggingface.co/bert-base-uncased
[5] https://huggingface.co/roberta-base

Table 1: Summary of datasets

Dataset	Total samples	Train	Test	Avg. sentence length	# Classes
MR	10,662	9K	1.06K	18.65	2
AGNews	127,600	120K	7.6K	38.57	4
Yelp	598,000	560K	38K	136.21	2
IMDB	50,000	25K	25K	227.14	2

4.2.1 AGNews

AGNews is a news categorization dataset constructed from the AG collection. AG is a collection of about one million news articles[6]. ComeToMyHead collected this dataset from more than 2,000 news sources in a period of over one year. The samples in AGNews are classified into four classes: business, world, sports, and science/technology. This dataset is split into training and test sets containing 120,000 and 7,600 samples, respectively. The dataset used in this work is available in Huggingface[7].

4.2.2 Rotten Tomatoes Movie Reviews (MR)

The MR dataset contains a total of 10,662 movie reviews, including 5,331 positive and 5,331 negative. The average length of the samples in the MR dataset is 32 words. The dataset is split into 8,530(80%) training, 1,066(10%) validation, and 1,066(10%) test samples. The dataset is publicly available in the Huggingface dataset repository[8].

4.2.3 Yelp Review Polarity Dataset

The Yelp review polarity dataset is extracted from data collected from the Yelp Dataset Challenge 2015. This dataset is constructed for binary sentiment classification. The reviews are labeled as positive or negative based on the rating scores provided by the reviewers. Scores of 1 and 2 are considered negative, while scores of 3 and 4 are positive reviews. The dataset is fully balanced, containing 280,000 training and 19,000 test samples for each polarity. The dataset can be accessed through Huggingface[9].

4.2.4 IMDB

The IMDB dataset consists of 50,000 labeled movie reviews collected from online sources. This dataset is divided into two halves, i.e., 25,000 training samples and 25,000 test samples. The average number of words per review sample is 215.63, which results in high computation time for adversarial attacks. Due to resource limitations, only 6,000 randomly selected samples are considered for the experiments. The dataset is available on Huggingface[10].

Each dataset is split into training, validation, and test data. The training data is used to train the target models and the test data is used to evaluate the model performance when attacked by the adversarial models. The test data is exposed to adversarial attacks and then passed through a system trained on the training data. This work exposes pre-trained models to the proposed attack recipe to report its effectiveness. Moreover, the training data is utilized during adversarial training to retrain the model with the actual training data augmented with adversarial samples generated for randomly

[6] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html
[7] https://huggingface.co/datasets/ag_news
[8] https://huggingface.co/datasets/rotten_tomatoes
[9] https://huggingface.co/datasets/yelp_polarity
[10] https://huggingface.co/datasets/imdb

selected samples from training data. The number of augmented samples varies depending on the user's choice.

4.3 Baseline Algorithms

Four well-known baseline algorithms are selected to compare the proposed model, including character-level, word-level, and multi-level perturbation. These recipes are briefly discussed as follows.

4.3.1 TextFooler

TextFooler [18] is a word-level, untargeted adversarial attack applied to both classification and entailment problems. TextFooler is a simple but strong baseline for generating utility-preserving adversarial examples in the black-box settings. This attack works on the principle of synonym substitution for the most important candidate words selected by the greedy-WIR method. In this attack, among all synonyms, only the ones with similar part-of-speech (POS) are considered as the initial candidates to avoid grammatical mistakes. Among these candidates, the ones with high universal sentence encoder (USE) scores are selected as the appropriate substitute to preserve the context of a sentence.

4.3.2 DeepWordBug

DeepWordBug [12] is a character-level adversarial attack that generates adversarial examples using four basic typographic editing operations: insertion, deletion, substitution, and transposition. In NLP, these operations may result in the most basic typographic spelling errors that may result in out-of-vocabulary words that are not recognized by the NLP algorithms without using spelling error correction techniques.

4.3.3 Pruthi

Pruthi et al. [13] is another character-level adversarial attack. It also uses composite transformation including insertion, deletion, transposition, and key-board-based transposition. Unlike DeepWordBug, which uses greedy-WIR, it uses a greedy algorithm for candidate selection.

4.3.4 TextBugger

TextBugger [11] is a multi-level adversarial attack that is reported to generate utility-preserving adversarial examples in both white-box and black-box environments. It introduces composite transformations to the most important words selected by the greedy-WIR method. These transformations include both word-level and character-level perturbations. The character-level perturbations used include insertion (space), deletion (random character), transposition (neighboring characters), and substitution (visually similar characters substitution), whereas word-level perturbation uses word substitution with the nearest neighbor selected from a set of top-k context-aware word vector space.

4.4 Evaluation Measures

The performance of the proposed attack is compared with the existing attacks based on the model effectiveness, efficiency, naturalness, human robustness, and utility robustness, where effectiveness and efficiency are quantitative, and naturalness and robustness are qualitative measures.

4.4.1 Effectiveness

Effectiveness is judged through the attack success rate and the model accuracy under attack.

- **Success rate:** The attack success rate is measured as the fraction of samples for which the attack is successful, i.e., that satisfies Eq. (2). It measures the effectiveness of the attack.
- **Accuracy under attack:** Accuracy under attack is the model accuracy over the adversarial samples. The accuracy under attack is also used to measure the effectiveness of the attack. Moreover, it can be used to measure the model's robustness against adversarial attacks.

4.4.2 Efficiency

The efficiency of the model is reported via the perturbation rate. The perturbation rate is measured as the fraction of words modified for a successful attack. An attack with a high success rate and a low perturbation rate is considered efficient.

4.4.3 Naturalness

Naturalness measures the closeness of the generated samples to social media text. Closeness is defined as the average similarity score of the adversarial samples to the social media text provided by the participants to the given adversarial samples. It provides evidence of the generated samples replicating the trends and patterns used in real-life scenarios.

4.4.4 Human Robustness

The human robustness metric is used to estimate the understanding and readability of the generated text by a human judge.

4.4.5 Utility Robustness

Utility robustness/preservation is the fraction of samples for which a human judge provides correct labels when provided with a mix of adversarial and legitimate samples.

5 Results and Evaluation

This section discusses the performance and comparison of the proposed model with the four state-of-the-art adversarial attack methods discussed in Section 4.3 on the target models introduced in Section 4.1. Moreover, the results yielded from human evaluation and ablation study are also discussed in Sections 5.2 and 5.4, respectively.

5.1 Results

This section discusses the proposed adversarial attack's effectiveness and efficiency compared to the existing techniques. The effectiveness of the proposed adversarial attack is evaluated based on the attack success rate and model accuracy under attack. The simulation results prove the effectiveness of the proposed attack over the existing baseline attacks, as shown in Tables 2 and 3.

Table 2: Comparison of the proposed model with the existing models in attack success rate

Model	Dataset	#Sample	TextFooler	Pruti.	DeepWordBug	TextBugger	Proposed
WordCNN	MR	1.06K	99.27	36.14	91.58	78.63	**99.88**
	AGNews	7.6K	98.88	11.25	98.84	86.15	**99.04**
	Yelp	8K	98.23	15.35	91.39	96.35	**98.97**
	IMDB	6K	99.01	30.96	86.72	98.60	**100**

(Continued)

Table 2 (continued)

Model	Dataset	#Sample	TextFooler	Pruti.	DeepWordBug	TextBugger	Proposed
Bi-LSTM	MR	1.06K	98.19	36.99	84.82	79.04	**99.16**
	AGNews	7.6K	94.93	9.79	87.87	75.45	**96.88**
	Yelp	8K	96.38	13.35	83.04	92.56	**98.43**
	IMDB	6K	98.90	27.16	77.63	94.61	**100**
BERT-uncased	MR	1.06K	89.31	41.2	77.17	62.25	**92.09**
	AGNews	7.6K	81.33	10.98	62.34	52.99	**85.42**
	Yelp	8K	94.20	11.93	68.89	83.35	**96.23**
	IMDB	6K	98.10	17.24	64.35	89.67	**99.27**
RoBERTa-base	MR	1.06K	93.76	37.53	80.13	66.91	**95.56**
	AGNews	7.6K	82.79	11.96	57.29	51.94	**84.58**
	Yelp	8K	90.70	11.55	64.62	78.94	**93.25**
	IMDB	6K	94.09	9.68	64.75	81.25	**97.36**

Table 3: Comparison of the proposed model with the existing models in model accuracy under attack

Model	Dataset	#Sample	Original	TextFooler	Pruti.	DeepWordBug	TextBugger	Proposed
WordCNN	MR	1.06K	76.83	0.56	49.06	6.47	16.42	**0.09**
	AGNews	7.6K	91.57	1.03	81.26	1.07	12.68	**0.88**
	Yelp	8K	88.29	1.56	74.74	7.60	3.23	**0.95**
	IMDB	6K	81.18	0.80	56.05	10.78	1.13	**0**
Bi-LSTM	MR	1.06K	77.86	1.41	49.06	11.82	16.32	**0.66**
	AGNews	7.6K	91.63	4.64	82.66	11.12	22.5	**2.86**
	Yelp	8K	91.4	3.31	79.2	15.50	6.8	**1.44**
	IMDB	6K	86.33	0.95	62.88	19.32	4.65	**0**
BERT uncased	MR	1.06K	84.24	9.01	49.53	19.23	31.8	**6.66**
	AGNews	7.6K	95.14	17.76	84.70	35.83	44.72	**13.87**
	Yelp	8K	97.20	5.64	85.60	30.24	16.19	**3.66**
	IMDB	6K	93.08	1.02	77.03	33.18	9.62	**0.68**
RoBERTa-base	MR	1.06K	88.70	5.53	55.44	17.64	29.36	**3.94**
	AGNews	7.6K	95.30	16.40	83.90	40.70	45.80	**14.70**
	Yelp	8K	97.80	9.10	86.50	34.60	20.60	**6.60**
	IMDB	6K	94.80	5.60	85.63	33.42	12.90	**2.50**

Table 2 shows the success rate of the proposed and existing attacks on the state-of-the-art text classification models using benchmark datasets in binary and multi-class classification. The success rate of the proposed attack is higher than the existing attacks with an improvement ranging from 0.16% to 4.09% over the best-performing among existing attacks (i.e., TextFooler).

Table 3 provides the evaluation and comparison of the models in terms of classification accuracy under attack. The proposed attack achieved high success in degrading the state-of-the-art model's accuracy. The highest degradation caused by the proposed attack is against the BERT base uncased on the IMDB dataset, which is from 93.08% to 0.68%. Surprisingly, the WordCNN and Bi-LSTM on IMDB dataset achieved the lowest accuracy of 0% on the adversarial samples.

The proposed model achieved the highest (100%) success rate on 6, 000 samples randomly selected from the IMDB dataset with modifications of only up to 2% of the words. WordCNN is proved to be the most vulnerable model to the proposed adversarial attacks, followed by Bi-LSTM. The BERT base uncased and RoBERTa base models showed robustness to a certain degree compared to the WordCNN and Bi-LSTM models. However, they are still vulnerable to simple perturbations.

Table 4 provides a comparison of the proposed attack with the existing attacks regarding the perturbation rate. An efficient adversarial attack is one that can achieve a high success rate with fewer perturbations. As shown in Tables 2 and 4, the simulation results demonstrate that the proposed model achieved a high success rate with relatively fewer perturbations.

Table 4: Comparison of the proposed model with the existing models in perturbation rate

Model	Dataset	# Words per Sample	TextFooler	Pruti.	DeepWordBug	TextBugger	Proposed
WordCNN	MR	18.65	13.83	7.61	18.03	27.23	**12.18**
	AGNews	38.57	15.17	2.91	21.07	57.14	**11.27**
	Yelp	136.21	5.77	2.45	8.68	52.36	**5.21**
	IMDB	227.14	2.21	2.32	3.42	28.21	**2.31**
Bi-LSTM	MR	18.65	12.98	7.74	16.81	25.12	**11.92**
	AGNews	38.57	17.36	2.91	20.63	57.86	**13.17**
	Yelp	136.21	6.63	2.39	8.62	49.84	**4.95**
	IMDB	227.14	2.30	2.08	4.06	43.59	**2.42**
BERT uncased	MR	18.65	19.40	8.64	21.00	15.28	**17.27**
	AGNews	38.57	23.35	5.32	25.12	33.86	**19.79**
	Yelp	136.21	10.50	7.21	11.71	25.06	**9.67**
	IMDB	227.14	9.04	10.21	6.24	28.21	**4.83**
RoBERTa-base	MR	18.65	18.44	8.22	21.78	18.13	**17.01**
	AGNews	38.57	23.33	3.03	24.84	32.87	**18.79**
	Yelp	136.21	10.50	7.01	11.52	25.73	**10.10**
	IMDB	227.14	9.21	10.05	6.24	26.92	**4.26**

Overall, the proposed attack and TextFooler performed better in degrading the model's performances. However, TextFooler is a word-level adversarial attack that is reported to have the limitation of replacing words with less frequently used words that are not necessarily known to other users. Moreover, word-level adversarial text generation is rarely used in real-life scenarios like social media communication. Among the low-level adversarial attacks, the DeepWordBug and TextBugger performed well compared to Pruthi. The adversarial samples generated by DeepWordBug contain OOV words, likely to be corrected (defended) by using spell-checkers. The samples generated by these

models are more similar to human-generated samples; however, it is not necessarily a close replication of the social media text. The human evaluation is discussed in the following Section 5.2.

5.2 Human Evaluation

This work conducted a study with human observers to estimate the quality of the adversarial samples in the context of naturalness, human robustness, and utility robustness. To measure naturalness, the participants were asked to score the generated adversarial samples from 1 to 5 based on their similarity with social media text. For human robustness, the participants were asked to identify the adversarial examples from a mix of adversarial and legitimate samples and suggest one or two correct replacements for the perturbed tokens. Moreover, the participants were asked to label the mix of adversarial and legitimate text from the given labels for utility-robustness analysis.

The participants were provided with a shuffled mix of 200 samples selected from both adversarial and original text using the MR dataset. The data consisted of 42% legitimate sentences and 58% adversarial samples. We asked 5 participants to evaluate the generated samples. We considered the average score provided by the participants for the generated text similarity with the social media text. For utility preservation, majority voting is used, i.e., the label provided by at least three of five participants was the given label. For human robustness estimation, three measures are considered: the average percentage of correctly identified adversarial and legitimate sentences by all participants, the percentage scores of each sentence correctly identified by the participants, and the percentage of correctly suggested words.

After examining the results, the average similarity score provided by the participants to the samples provided was 4.6 out of 5. Interestingly, the adversarial examples generated were scored higher than the legitimate samples. The average similarity score of the generated samples was 4.85, and legitimate samples scored 4.35. The high similarity score of the adversarial samples results from our intention to replicate the most common textual variations found in online conversations.

About 96.01% of the legitimate sentences and 94.2% of the adversarial samples were labeled correctly by the participants. The participants were asked to label the sentences as positive and negative. The labeling of sentences was independent of the identification of adversarial and legitimate sentence identification. Some users identified some legitimate sentences as adversarial; however, they labeled them correctly, and vice versa.

The participants identified, on average, 67.45% of the total sentences correctly as legitimate or adversarial. Individually, 82.21% of the legitimate and 52.69% of the adversarial sentences were identified correctly. The average individual score of the sentences correctly identified was 4.01, i.e., four people correctly identified each sentence on average.

Additionally, the participants were asked to provide suitable candidates for the samples, which they marked as adversarial examples. A total of 61 sentences out of 116 were classified correctly as adversarial samples by three or more participants, and 95% of the total suggestions consisted of single words. The participants correctly provided 89.02% of the suggestions for the sentences, which were correctly identified as adversarial samples. The suggestions provided by the participants to the samples correctly classified as adversarial by all participants were 100% correct.

For comparison of the proposed model with existing approaches, the participants were provided with a mix of original and adversarial samples generated by the DeepWordBug, TextBugger, and Eger et al. [32]. Their task was to classify them into their respective classes, identify whether these samples are adversarial or original, and determine their similarity to human-generated text. The

participants could correctly label up to 91% of the sentences, with DeepWordBug achieving the highest score of 96% and Eger et al. [32] having the lowest score of 80%. The participants, however, correctly identified all of the adversarial samples generated by Eger et al. [32] because out of nine transformation methods, the inner-shuffle, full-shuffle, intrude, and visual methods were used in this work. Unlike the other method, these methods have less similarity to human-generated text and hence can be easily identified as adversarial by human participants. The participants correctly identified 65% of the samples generated by the DeepWordBug, and 77% of the adversarial samples generated by the TextBugger method.

5.3 Examples of Generated Adversarial Sentences

Table 5 shows some examples of the adversarial samples generated by the existing and proposed methods. As seen in Table 5, the samples generated by the proposed method are consistent with the original input while successfully deceiving the classification model. By manually evaluating random samples, it is observed that although the word-level and multi-level adversarial perturbation performed better, they replaced simple words with complex and less frequently used words, i.e., "consistently → ceaselessly" and "artistes → virtuoso," etc. Some words inserted by these attacks did not fit the following words, such as "less funny → cheaper funny," etc. The character-level attacks, however, were more readable to humans and were closer to the text generated in informal communication, as shown in Table 5.

Table 5: Some examples of adversarial samples

Type	Sample
Polarity	Positive → Negative
Original	Lovingly photographed in the manner of a golden book sprung to life, stuart little 2 manages sweetness largely without stickiness.
TextFooler	Lovingly photographed in the manner of a golden book sprung to life, stuart little 2 administration sweetness largely without stickiness.
DeepWordBug	Lovingly photographed in the manner of a golden book sprung to life, stuart little 2 manages sweetness largely iwthout scickiness.
TextBugger	Lovingly photographed in the manner of a golden book sprung to life, stuart little 2 manages sweetness largely without ??tickiness.
Proposed	Lovingly photographed in the manner of a golden book sprung to life, stuart little 2 manages sweetness largely wthout stickiness.
Polarity	Positive → Negative
Original	Consistently clever and suspenseful.
TextFooler	Ceaselessly cleverer and enigmatic.
DeepWordBug	Consistently celver and thuspneseful.
TextBugger	Consistentl? clever and suspenseful.
Proposed	Cnsistntly clever and suspenseful.
Polarity	Negative → Positive
Original	Less funny than it should be and less funny than it thinks it is.
TextFooler	Down funny than it should be and cheaper funny than it thinks it is.
DeepWordBug	Resin funny than it should be and Wes funny than it thinks it is.

(Continued)

Table 5 (continued)

Type	Sample
TextBugger	Least funny than it should be and lass funny than it thinks it is.
Proposed	Less fnny than it should be and l3ss funny than it thinks it is.
Polarity	Negative → Positive
Original	The cinematic equivalent of patronizing a bar favored by pretentious, untalented artistes who enjoy moaning about their cruel fate.
TextFooler	The cinematic equivalent of haughty a counsel favored by ostentatious, untalented virtuoso who enjoy whinging about their cruel fate.
DeepWordBug	The I/cinematic equivalent of ptronizing a br favored by pretentiLus, untalented arristes who enjoy moaing about their cruel fate.
TextBugger	The cinematic equivalent of patronizig a br favored by ostentatious, untalented artistes who enjoy moainng about their cruel fate.
Proposed	The cinematic equv1nt of ptrnz#ing a bar favored by pre10tious, untalented artistes who enjoy moaning about their cruel fate.
Polarity	Negative → Positive
Original	Koepp's screenplay isn't nearly surprising or clever enough to sustain a reasonable degree of suspense on its own.
TextFooler	Koepp's screenplay isn't almost stun or Plan enough to sustain a prudent degree of suspense on its own.
DeepWordBug	Koepp's screepnlay isn't early surprisig or &leer Uenouhl to sustain a reasonable degree of suspense on its own.
TextBugger	Koepp's screenplay isn't nearly staggering or celver enoIgh to sustain a reasonable degree of suspense on its own.
Proposed	Koepp's screenplay isn't nearly srprsing or clever engh to susta-in a reasonable degree of suspense on its own.

5.4 Ablation Study

The proposed method uses composite transformation, where one or more types of perturbations are introduced to generate adversarial samples. Therefore, an ablation study is conducted to determine how the proposed method performs when these transformations are individually applied to attack the classification model. Moreover, the performance of the proposed attack with other search mechanisms like genetic algorithms and beam search methods is also examined.

The average perturbation ratio of different types of datasets (MR and AGNews) against BERT base uncased is shown in Fig. 1. The vowel removal is the most effective type of perturbation. In this experiment, the data was attacked by using the proposed composite transformation.

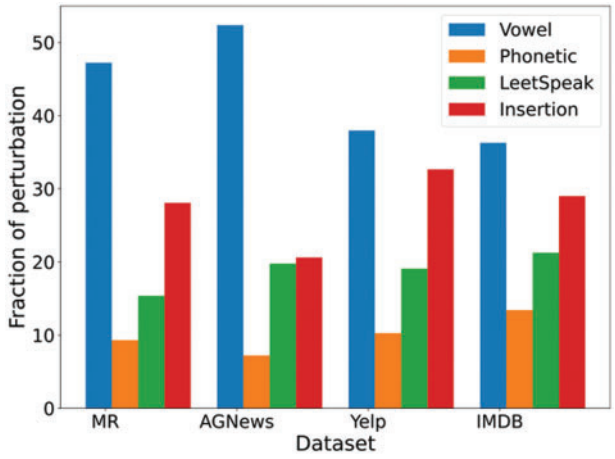

Figure 1: Fraction of perturbations against BERT base uncased model

Fig. 2 shows the performance of the individual transformations against the BERT base uncased model on the MR dataset. It is observed from simulations of individual transformations that vowel removal was the most effective attack, followed by character insertion and LeetSpeak. The vowel removal however consists of rule-based vowel removal and shwa removal. It is found that among the phonetic-based perturbations, the double meta-phone performed better than the rule-based mapping of words and sub-words to similar-sounding characters; however, the performance of such rule-based phonetic transformations was comparable.

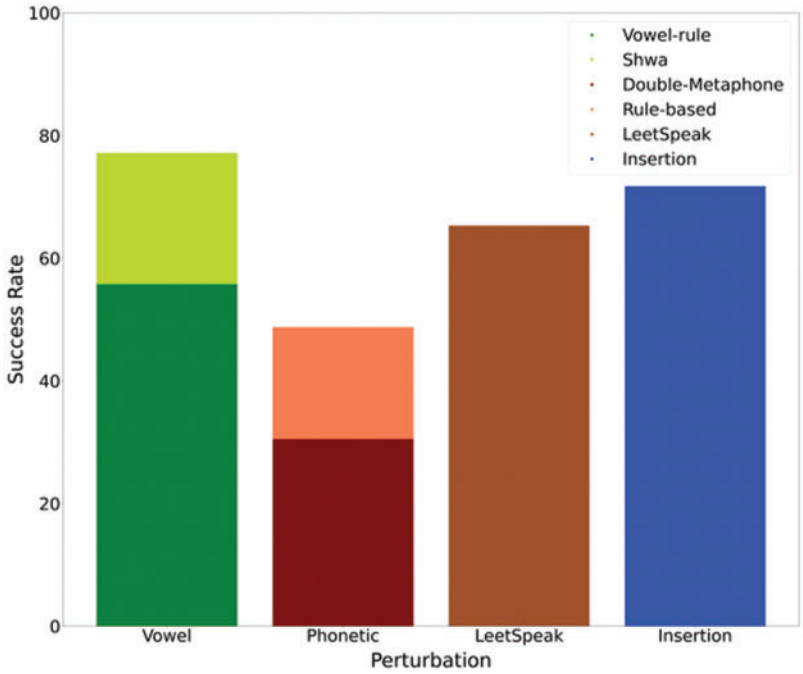

Figure 2: Performance of individual transformations against BERT base uncased on MR dataset

Table 6 shows the proposed method's performance compared to the other search algorithms. The simulation results showed that the performance of the proposed word importance weight ranking-based beam search method was the best among the selected search methods, followed by the beam search algorithm. The beam search method with the proposed transformations performs well compared to the existing transformations, as evident in Table 6. The word importance ranking improved the performance of the beam search-based adversarial perturbations.

Table 6: Effectiveness of the proposed word-importance ranking-based beam search algorithm in comparison to existing searching mechanisms

Model	Dataset	Samples	Original	GS		BS		Proposed	
			Accuracy	Success rate	Accuracy	Success rate	Accuracy	Success rate	Accuracy
WordCNN	MR	1.06K	76.83	97.44	1.97	99.15	0.66	99.88	0.09
	AGNews	7.6K	91.57	95.23	4.37	98.92	0.99	99.04	0.88
	Yelp	8K	88.29	96.29	3.28	98.37	1.44	98.92	0.95
	IMDB	6K	81.18	98.21	1.45	99.06	0.77	100	0
Bi-LSTM	MR	1.06K	77.86	96.75	2.53	98.55	1.13	99.16	0.66
	AGNews	7.6K	91.63	92.23	7.12	95.39	4.22	96.88	2.86
	Yelp	8K	91.4	94.79	4.76	96.61	3.10	98.43	1.44
	IMDB	6K	86.33	97.18	2.43	99.02	0.85	100	0
Bert-uncased	MR	1.06K	84.24	85.08	12.57	90.65	7.88	92.09	6.66
	AGNews	7.6K	95.14	77.69	21.22	82.53	16.62	85.42	13.87
	Yelp	8K	97.2	93.92	5.91	95.31	4.56	96.23	3.66
	IMDB	6K	93.08	95.72	3.98	98.98	0.95	99.27	0.68
RoBERTa-base	MR	1.06K	88.70	77.56	19.90	90.53	8.40	95.56	3.94
	AGNews	7.6K	95.30	69.25	29.30	80.85	18.25	84.58	14.70
	Yelp	8K	97.80	88.46	11.29	91.97	7.85	93.25	6.60
	IMDB	6K	94.80	87.24	12.10	92.72	6.90	97.36	2.50

6 Defense Mechanisms against Adversarial Attacks

This section reports on experiments with two of the most commonly used defense techniques, text normalization, and adversarial training, to defend against the proposed adversarial attack.

6.1 Adversarial Training (AT)

To guard against adversarial attacks, one needs to train a classifier F that can guarantee both $F(x) = y_{true}$ and $F(x') = y'_{true}$ with high confidence scores. Adversarial training (AT) is one of the most effective methods to enhance a model's robustness [3]. AT is the process of training a model on the data augmented with adversarial examples. Adversarial examples are label-preserved modified versions of the actual samples in a dataset. Therefore, for a class in a classification problem, instead of only the actual textual sequence, the model will be able to learn from the perturbed samples as well. The proportion of modified data augmentation may vary depending on the experimental settings,

application, and the desirable robustness level with the available resources. In this work, the target models were trained with the augmented datasets for three epochs using a learning rate of 0.00005.

Table 7 shows the performance of the AT with the proposed model. The actual accuracy is the accuracy of the model with legitimate training and test samples, the adversarial accuracy is the prediction accuracy of the new model trained with the augmented dataset over the legitimate test set, and the success rate (SR) is the success of the adversarial attack on the actual model, and the SR-with-AT is the proportion of misclassified samples by the new model trained with the augmented dataset. Table 7 shows that adversarial training increases the model's robustness against adversarial attacks with a slight trade-off in the model's performance. For instance, the attack success rate against BERT on the MR dataset is decreased by about 60% with a performance degradation of about 1%. Therefore, better adversarial training can be considered a potential solution to the proposed attack.

Table 7: Performance of the DL models trained with legitimate and adversarial samples generated by the proposed attack

Dataset	Model	Actual samples	Adv. samples	Actual accuracy	Adv. accuracy	Success rate under attack	Success rate with AT
MR	WordCNN	9K	3K	76.83	73.49	99.88	56.72
	Bi-LSTM	9K	3K	77.86	75.33	99.16	53.19
	BERT-uncased	9K	3K	84.24	83.01	92.09	32.12
	RoBERTa-base	9K	3K	88.70	84.29	95.56	38.72
AGNews	WordCNN	120K	3K	91.57	87.21	99.04	58.29
	Bi-LSTM	120K	3K	91.63	88.92	96.88	57.73
	BERT-uncased	120K	3K	95.14	93.37	85.42	29.98
	RoBERTa-base	120K	3K	95.30	94.47	84.58	34.13

Despite its potential, the practical implementation of adversarial training is limited by its dependency on the type and number of adversarial samples. This requires sufficient knowledge about the attack strategy; however, an attacker does not announce the details of an attack.

6.2 Text Normalization

The adversarial attack proposed in this work fools the target models by modifying certain words in the input samples at the character level. The adversarial samples generated by this recipe contain OOV words, which causes the target model to generate the wrong output because the model is unaware of the OOV text. One of the ways to defend against such an attack is adversarial training, as discussed in the previous Section 6.1, in which the model is trained on both legitimate and adversarial samples so that the model is familiar with both the legitimate and adversarial samples.

Text normalization is the process of translating noisy and non-standard OOV words into their standard lexical representation. The spelling error correction algorithm is one of the most basic and straightforward text normalization methods, where the OOV words are considered misspelled words.

These algorithms normalize noisy misspelled OOV words to their standard lexical counterparts by using spell-checking algorithms to enhance the performance of the traditional text analysis methods [43,44]. Spell-checkers are reported in the literature [11,13] to perform well in defense against character-level adversarial attacks. Text normalization is not a "one size fits all" task of substituting OOV words with their valid replacements [45]. Besides the correction of the misspelled words, a normalization algorithm needs to handle a wide range of OOV words by sensing the error patterns, identifying the error types, and activating the appropriate correction methods.

In this work, a modified version of [46] is applied to detect and correct the OOV words in the input sequence that may be generated due to an adversarial attack before passing it through the classification model. The text normalization method is a stacked ensembled-based text normalization that utilizes a dictionary lookup method for OOV word detection and a hybrid method for OOV word correction. This method's architectural and implementation details are discussed in [46].

Table 8 shows the effectiveness of the text normalization method in defense against character-level adversarial attacks. The simulation results showed that text normalization could be an effective and generalized defense against character-level adversarial attacks. The text normalization method is an extension of spell correction approaches suggested to defend against low-level attacks. The spell correction methods, however, fail to correct multiple perturbations and may not be very effective. The traditional spell correction method is a suitable defense against adversarial attacks, which may result in common typing mistakes, i.e., DeepWordBug [12], and Pruthi et al. [13]. We employed the text normalization as a defense to the character-level adversarial attacks to provide evidence of its effectiveness. The proposed text normalization method is very effective against the DeepWordBug method, due to its simple basic character-level editing operations. Similarly, in the case of TextBugger at the character level perturbations, the normalization was an effective defense, however, in case of word-level perturbations the text normalization failed to identify and defend the adversarial samples.

Table 8: Effectiveness of the text normalization method in defense against adversarial attacks

Dataset	Model	DeepWordBug success rate		TextBugger success rate		Proposed success rate	
		Without normalization	With normalization	Without normalization	With normalization	Without normalization	With normalization
MR	WordCNN	91.58	14.77	78.63	18.29	99.88	37.32
	Bi-LSTM	84.82	11.28	79.04	17.77	99.16	33.03
	BERT-uncased	77.17	8.93	62.25	10.69	92.09	22.62
	RoBERTa-base	80.13	5.31	66.91	8.22	95.56	23.18
AGNews	WordCNN	98.84	13.63	86.15	15.18	99.04	35.31
	Bi-LSTM	87.87	9.79	75.45	14.42	96.88	32.83
	BERT-uncased	62.34	7.22	52.99	10.94	85.42	20.19
	RoBERTa-base	57.29	3.78	51.94	7.87	84.58	18.79

The text normalization was an effective defense against the proposed adversarial attack, which provide a way to utilize more advanced text normalization methods, that can handle multiple complex character level perturbations simultaneously, as a defense mechanism against the adversarial attacks.

7 Conclusion

This work focused on generating adversarial attacks inspired by the human cognition of text generation in online social media conversations. We selected four simple and most prominently used

textual variations introduced intentionally in online communication. This paper introduced a word importance ranking-based beam search algorithm in the proposed attack to increase its effectiveness.

We studied adversarial attacks against state-of-the-art text classification models in untargeted black-box settings, including WordCNN, Bi-LSTM, BERT, and RoBERTa. Extensive simulations demonstrated the effectiveness and efficiency of the proposed adversarial attack over the most relevant existing attacks in all cases studied in this work. The adversarial samples generated by the proposed attack were more natural and similar to the text generated on social media platforms, as evaluated by human participants. The adversarial examples preserve human and utility robustness.

Moreover, this article presented the effectiveness of two potential defenses against such adversarial attacks, i.e., adversarial training and text normalization. It is observed that the attack success rate was reduced by about 39% at minimum when using 3,000 adversarial samples augmented with legitimate training data. This article presented a normalization method that converts the input text to standard lexical form before passing it to the classification model. The proposed normalization method in cascade with transformer-based sequence-to-sequence substitution can effectively defend against character-level adversarial attacks. Despite the high effectiveness and closeness to real-life social media text, the proposed attack can be improved by using automatic methods to learn human cognition in social media text generation, i.e., automatically learn transformations from social media text. Developing a generic defense mechanism for character-level adversarial attacks has potential for future work.

Acknowledgement: None.

Funding Statement: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(No. NRF-2022R1A2C1007434), and also by the BK21 FOUR Program of the NRF of Korea funded by the Ministry of Education (NRF5199991014091).

Author Contributions: All authors contributed equal to this work.

Availability of Data and Materials: The data used in this work is publicly available. All data that are not public are available from the corresponding author on request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Said, K. Ahmad, M. Riegler, K. Pogorelov, L. Hassan *et al.,* "Natural disasters detection in social media and satellite imagery: A survey," *Multimedia Tools and Applications*, vol. 78, no. 22, pp. 31267–31302, 2019.

[2] I. Alsmadi, K. Ahmad, M. Nazzal, F. Alam, A. Al-Fuqaha *et al.,* "Adversarial NLP for social network applications: Attacks, defenses, and research directions," *IEEE Transactions on Computational Social Systems*, pp. 1–20, 2022.

[3] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:*1412.6572, 2014.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan *et al.,* "Intriguing properties of neural networks," in *2nd Int. Conf. on Learning Representations (ICLR)*, Banff, AB, Canada, 2014.

[5] S. Eger, G. G. Şahin, A. Rücklé, J. U. Lee, C. Schulz *et al.,* "Text processing like humans do: Visually attacking and shielding NLP systems," in *Proc. of the 2019 Conf. of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, USA, vol. 1, pp. 1634–1647, 2019.

[6] S. Feng, E. Wallace, A. Grissom II, P. Rodriguez, M. Iyyer *et al.,* "Pathologies of neural models make interpretations difficult," in *Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 3719–3728, 2018.

[7] M. Alzantot, Y. S. Sharma, A. Elgohary, B. J. Ho, M. Srivastava *et al.,* "Generating natural language adversarial examples," in *Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 2890–2896, 2018.

[8] M. T. Ribeiro, S. Singh and C. Guestrin, "Semantically equivalent adversarial rules for debugging NLP models," in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, vol. 1, pp. 856–865, 2018.

[9] B. Liang, H. Li, M. Su, P. Bian, X. Li *et al.,* "Deep text classification can be fooled," in *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence*, Stockholm, Sweden, pp. 4208–4215, 2017.

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

[11] J. Li, S. Ji, T. Du, B. Li and T. Wang, "Textbugger: Generating adversarial text against realworld applications," in *26th Annual Network and Distributed System Security Symp.*, San Diego, CA, USA, 2019.

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

[13] D. Pruthi, B. Dhingra and Z. C. Lipton, "Combating adversarial misspellings with robust word recognition," in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 5582–5591, 2019.

[14] W. E. Zhang, Q. Z. Sheng, A. Alhazmi and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 3, pp. 1–41, 2020.

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

[16] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 2021–2031, 2017.

[17] M. T. Ribeiro, T. Wu, C. Guestrin and S. Singh, "Beyond accuracy: Behavioral testing of NLP models with CheckList," in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, 2020.

[18] D. Jin, Z. Jin, J. T. Zhou and P. Szolovits, "Is bert really robust? A strong baseline for natural language attack on text classification and entailment," in *Proc. of the AAAI Conf. on Artificial Intelligence*, New York, NY, USA, vol. 34, pp. 8018–8025, 2020.

[19] S. Ren, Y. Deng, K. He and W. Che, "Generating natural language adversarial examples through probability weighted word saliency," in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 1085–1097, 2019.

[20] Y. Zang, F. Qi, C. Yang, Z. Liu, M. Zhang *et al.,* "Word-level textual adversarial attacking as combinatorial optimization," in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6066–6080, 2020.

[21] L. Li, R. Ma, Q. Guo, X. Xue and X. Qiu, "BERT-ATTACK: Adversarial attack against BERT using BERT," in *Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6193–6202, 2020.

[22] D. Li, Y. Zhang, H. Peng, L. Chen, C. Brockett *et al.,* "Contextualized perturbation for textual adversarial attack," in *Proc. of the 2021 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5053–5069, 2021.

[23] X. Wang, Y. Yang, Y. Deng and K. He, "Adversarial training with fast gradient projection method against synonym substitution based text attacks," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 35, pp. 13997–14005, 2021.
[24] H. Gao, H. Zhang, X. Yang, W. Li, F. Gao *et al.,* "Generating natural adversarial examples with universal perturbations for text classification," *Neurocomputing*, vol. 471, pp. 175–182, 2022.
[25] N. Zhou, N. Yao, J. Zhao and Y. Zhang, "Rule-based adversarial sample generation for text classification," *Neural Computing and Applications*, vol. 34, no. 13, pp. 10575–10586, 2022.
[26] J. Ebrahimi, A. Rao, D. Lowd and D. Dou, "Hotflip: White-box adversarial examples for text classification," in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, vol. 2, pp. 31–36, 2018.
[27] J. Ebrahimi, D. Lowd and D. Dou, "On adversarial examples for character-level neural machine translation," in *Proc. of the 27th Int. Conf. on Computational Linguistics*, Santa Fe, New Mexico, USA, pp. 653–663, 2018.
[28] Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," in *Int. Conf. on Learning Representations*, Vancouver, BC, Canada, 2018.
[29] H. Hosseini, S. Kannan, B. Zhang and R. Poovendran, "Deceiving Google's perspective API built for detecting toxic comments," *arXiv preprint arXiv:*1702. 08138, 2017.
[30] N. Rodriguez and S. Rojas-Galeano, "Shielding Google's language toxicity model against adversarial attacks," *arXiv preprint arXiv:*1801.01828, 2018.
[31] S. Tan, S. Joty, M. Y. Kan and R. Socher, "It's morphin'time! combating linguistic discrimination with inflectional perturbations," in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2920–2935, 2020.
[32] S. Eger and Y. Benz, "From hero to zéroe: A benchmark of low-level adversarial attacks," in *Proc. of the 1st Conf. of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th Int. Joint Conf. on Natural Language Processing*, Suzhou, China, pp. 786–803, 2020.
[33] R. Bhalerao, M. Al-Rubaie, A. Bhaskar and I. Markov, "Data-driven mitigation of adversarial text perturbation," *arXiv preprint arXiv:*2202.09483, 2022.
[34] T. Le, J. Lee, K. Yen, Y. Hu and D. Lee, "Perturbations in the wild: Leveraging human-written text perturbations for realistic adversarial attack and defense," in *Findings of the Association for Computational Linguistics*, Dublin, Ireland, pp. 2953–2965, 2022.
[35] T. Niven and H. Y. Kao, "Probing neural network comprehension of natural language arguments," in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 4658–4664, 2019.
[36] D. Boyd, S. Golder and G. Lotan, "Tweet, tweet, retweet: Conversational aspects of retweeting on Twitter," in *Proc. of the 2010 43rd Hawaii Int. Conf. on System Sciences*, Honolulu, HI, USA, pp. 1–10, 2010.
[37] C. Tagg, "Wot did he say or could u not c him 4 dust? Written and spoken creativity in text messaging," *Transforming Literacies and Language: Multimodality and Literacy in the New Media Age*, vol. 223, pp. 223–236, 2011.
[38] L. Philips, "The double metaphone search algorithm," *C/C++ Users Journal*, vol. 18, no. 6, pp. 38–43, 2000.
[39] J. Y. Yoo, J. Morris, E. Lifland and Y. Qi, "Searching for a search method: Benchmarking search algorithms for generating nlp adversarial examples," in *Proc. of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 323–332, 2020.
[40] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin *et al.,* "TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP," in *Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 119–126, 2020.
[41] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, USA, vol. 1, pp. 02, 2019.

[42] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi *et al.,* "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:* 1907.11692, 2019.
[43] J. Khan and S. Lee, "Enhancement of sentiment analysis by utilizing noisy social media texts," *The Journal of Korean Institute of Communication Sciences*, vol. 45, no. 6, pp. 1027–1037, 2020.
[44] A. Lertpiya, T. Chalothorn and E. Chuangsuwanich, "Thai spelling correction and word normalization on social text using a two-stage pipeline with neural contextual attention," *IEEE Access*, vol. 8, pp. 133403–133419, 2020.
[45] T. Baldwin and Y. Li, "An in-depth analysis of the effect of text normalization in social media," in *Proc. of the 2015 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, USA, pp. 420–429, 2015.
[46] J. Khan and S. Lee, "Enhancement of text analysis using context-aware normalization of social media informal text," *Applied Sciences*, vol. 11, no. 17, pp. 8172, 2021.