Neurocomputing 500 (2022) 135–142

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Leveraging transferability and improved beam search in textual adversarial attacks

Bin Zhu [a], Zhaoquan Gu [a,b,*], Yaguan Qian [c], Francis Lau [d], Zhihong Tian [a,b]

[a] Guangzhou University, Guangzhou 510006, China
[b] Department of New Networks, Peng Cheng Laboratory, Shenzhen 999077, China
[c] Zhejiang University of Science and Technology, Hangzhou 310023, China
[d] The University of Hong Kong, Hong Kong 999077, China

ARTICLE INFO

Article history:
Received 17 November 2021
Revised 23 March 2022
Accepted 14 May 2022
Available online 18 May 2022
Communicated by Zidong Wang

Keywords:
Deep neural networks
Natural language processing
Adversarial examples
Textual attacks

ABSTRACT

Adversarial attacks in NLP are difficult to ward off because of the discrete and highly abstract nature of human languages. Prior works utilize different word replacement strategies to generate semantic-preserving adversarial texts. These query-based methods, however, have limited exploration of the search space. To fully explore the search space, an improved beam search with multiple random perturbing positions is used. Besides, we use the transferable vulnerability from surrogate models to choose vulnerable candidate words for target models. We empirically show that beam search with multiple random attacking positions works better than the commonly used greedy search with word importance ranking. Extensive experiments on three popular datasets demonstrate that our method can outperform three advanced attacking methods under black-box settings. We provide ablation studies to clearly show the effectiveness of our improved beam search which can achieve a higher success rate than the greedy approach under the same query budget.

1. Introduction

Deep neural networks (DNNs) have achieved results that exceed human levels in many tasks in the natural language processing (NLP) field [1–3], but the predictions of models used in these tasks cannot be explained in a way that humans can understand, revealing these models' lack of interpretability [4]. In connection with this, recent studies have shown that DNNs can be easily deceived to make wrong decisions, by crafted adversarial examples [5,6], which is one of the major vulnerabilities of DNNs.

In this paper, we focus on adversarial attacks under black-box settings—i.e., any access to the model is limited to the input and output. The study of black-box attacks is more challenging and realistic [7] than non-black-box ones.

In the NLP field, existing methods of black-box adversarial attacks can be roughly divided into two categories: 1) *Query-based attacks* find vulnerable tokens by querying the output decisions and scores of the target model, and then apply different strategies to these tokens to generate adversarial texts; 2) *transfer-based attacks* utilize a surrogate model to approximate the decision boundary of the target model and perform gradient-based attacks on the surrogate model. Since adversarial examples are proven to be transferable [8–10], these adversarial examples can also fool the target model.

These existing methods however have limitations. On the one hand, query-based attacks require a large number of queries for the target model. Besides, unlike continuous data, the discrete nature of tokens makes it practically impossible to apply optimization algorithms to the search for vulnerabilities. Thus the vulnerable tokens can only be found by exhaustive search [11]. On the other hand, transfer-based attacks are subject to the transferability of the generated adversarial examples and could maintain only a low success rate.

There are also difficulties in textual adversarial attacks themselves. First, note that text data is discrete. Usually, we vectorize the tokens as the input of DNNs. Applying gradient-based adversarial attacks on these vectors may result in invalid and out of vocabulary tokens in the generated adversarial examples [12–14]. Second, the change of just one character or word may cause semantic changes and grammatical errors, which can be easily detected.

To tackle the above problems, we combine the advantages of these two methods. By applying the query-based search method on the word embeddings of the surrogate model, we use the combination of target model's prediction scores and semantic similarity as the objective function to efficiently generate textual adversarial examples.

* Corresponding author.
E-mail address: zqgu@gzhu.edu.cn (Z. Gu).

https://doi.org/10.1016/j.neucom.2022.05.054
0925-2312/

Our contributions are summarized as follows:

(1) We propose a black-box adversarial attack method that leverages an improved beam search and transferability from surrogate models, which can efficiently generate semantic-preserved adversarial texts.
(2) We evaluate our method on three popular datasets and four neural networks. Our method outperforms three advanced methods in automatic evaluation.
(3) We provide extensive ablation studies on the substitute strategy, revealing that beam search with multiple random attacking positions works better than the commonly used greedy search with word importance ranking.
(4) We make a thorough evaluation of the trade-off between query budgets and attack performance and provide an in-depth analysis of our method.

2. Background and Related Work

2.1. Textual Adversarial Examples

In this paper, we discuss adversarial examples created by deliberately adding minor perturbations to the clean examples. Formally, for a trained classifier F, a clean sample X, the ground truth label Y_{true}, and the maximum allowed perturbation ϵ, the adversarial example X_{adv} can be described as

$$F(X_{adv}) \neq Y_{true}, \|X - X_{adv}\| \leqslant \epsilon, \tag{1}$$

which is a non-targeted attack. Moreover, when the output of F is misled to be a specified label c, i.e., $F(X_{adv}) = c$, it is called a targeted attack.

Clean sample X is composed of a series of tokens, expressed as $X = [w_1, w_2, \ldots, w_n]$, where n is the number of tokens. In the black-box setting, the generation of textual adversarial examples can be considered a search problem. The fragility of the model is discovered through a large number of queries. The crafted X_{adv} cause the model to misclassify while keeping the perturbation at a minimum, which can be described as

$$\begin{aligned} X_{adv} &= \text{argmin}_{X_{adv}} \|X - X_{adv}\|, \\ &s.t.\ F(X_{adv}) \neq Y_{true}. \end{aligned} \tag{2}$$

The major issue then is how to measure the distance between X and X_{adv} so that the change in semantics can be well approximated. Commonly used methods include Word Mover's Distance (WMD) [15], cosine similarity and Euclidean distance in the embedding space [16,17], etc. Recent works have shown effective ways to maintain grammatical correctness and semantic similarity by using synonym and sememe substitutions [18–21].

2.2. Black-box Textual Adversarial Attacks

For discrete data like texts, it is hard to apply gradient-based methods directly. Since a sentence is composed of ordered tokens, discrete operations such as addition, deletion and replacement can be used to generate adversarial texts in an intuitive way. Essentially, these operations are applied in searches for adversarial texts. Under the black-box setting, search on discrete data can be regarded as a combinatorial optimization problem. Many black-box word replacement strategies are carefully designed to reduce the size of the search space in order to speed up the search.

[22] generate adversarial examples by randomly deleting or adding meaningless characters or numbers in the original texts, but that might corrupt the readability of the sentences, introduce grammatical errors, and even change the original meaning. These random methods essentially only explore a tiny search space and therefore can claim only a low attack success rate. [16] use a population-based optimization algorithm to reorganize the original sentence, yielding sentences with good attack effects. Heuristic algorithms [23,24] could increase the efficiency while expanding the search space, but the improvement is not significant. [18] meticulously design the Probability Weighted Word Saliency (PWWS) method to find the vulnerability tokens in the original sentence and use the strategy of synonym replacement to reduce the search space. [19] follow the word saliency method and add more constraints in the replacement procedure, such as part-of-speech (POS) filtering and semantic checking, in order to improve the semantic similarity of adversarial examples and reduce grammatical errors. [20] apply a particle swarm algorithm and a sememe replacement strategy to the search of adversarial examples. [21,25] use Masked Language Model (MLM) to generate candidate words to ensure that the original grammar and semantics are not broken..

Besides, there are works [26,8] that train surrogate models to estimate the decision boundary of the target model and apply gradient-based methods on the surrogate model to generate adversarial examples to attack the target model. Subject to the estimation error of the decision boundary, the success rate of these methods is generally not high.

Among the related works we could find many excellent methods for black-box adversarial example generation. However, much of them are based on greedy algorithms [7,18,19], which tend to apply insufficient exploration to the search space, and these methods have a not so strict constraint of semantic similarity, thus leaving much room for further improvement. Recently, [27] proposed an improved beam search algorithm to generate adversarial examples, which is also effective.

3. Methodology

In this section we explain each part of our algorithm in detail. We show our proposed attack in Algorithm 1.

Algorithm 1: Textual Adversarial Attack

Input: The original text X, ground truth label Y_{true}, classifier F, word count n of X, max modification percentage m, semantic similarity function $Sim(\cdot)$, similarity threshold θ_{USE}, beam size b
Output: The adversarial text X_{adv}
Initialization: $X_{adv} \leftarrow X, Adv\_list \leftarrow []$ // python list
Append X_{adv} in Adv_list
for i = 1 **to** $n \cdot m$ **do**
  for each sentence s_j in Adv_list **do**
    Compute integrated gradient G_s via Eq. (3)
    Randomly select N words in s_j
    for each word w_k in the selected words **do**
      Get candidate set $\mathbb{V}_{w_k}$ and choose s candidate words to substitute
    end for
    Keep all the modified sentences
  end for
  if there exist sentences satisfying $F(X_{adv}) \neq Y_{true}$ and $Sim(X, X_{adv}) \geqslant \theta_{USE}$ **then**
    Select the sentence with the largest semantic similarity as X_{adv}
    break
  else
    Clear Adv_list and append b sentences with the highest score via Eq. (7)
  end if
end for
return X_{adv}

3.1. Search Method

3.1.1. Searching in the Embedding Space

Optimization methods on continuous data cannot be directly applied to discrete tokens, so searching in the embedding space appears to be the only option. Given a sentence X, which would be vectorized and used as the input to the neural network. Denote the sentence embedding of X as $E_X = [E_1, E_2, \ldots, E_n]$, where E_i is the word embedding of the i^{th} token, which can be expressed as a d-dimensional vector $E_i = [e_1, e_2, \ldots, e_d]$. In order to perform adversarial attacks, a minor perturbation $\epsilon = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n]$ is added to E_X, so that the model would end up misclassifying while keeping the perturbation imperceptible.

In the high-dimensional embedding space, random search without direction is inefficient. Therefore, we use the gradient of the surrogate model to approximate the gradient of the target model. Given the surrogate model S, the gradient G_s w.r.t. the clean sample X is

$$G_s = \nabla_X \mathscr{L}(X, Y_{true}, \Theta), \tag{3}$$

where $\mathscr{L}$ is the loss function, Θ is the parameter of the surrogate model. The normalized gradient g_s is taken as the direction of the perturbation.

The intuition here is that even if the decision boundary of the surrogate model does not closely approximate that of the target model, due to the inherent transferability of vulnerabilities, there is still a high probability that the target model is vulnerable to these perturbations. The iterative procedure of searching for the optimal representation that maximizes the loss function can be expressed as

$$E_{X^{(t+1)}} = E_{X^{(t)}} + N\left(\nabla_{X^{(t)}} \mathscr{L}\left(X^{(t)}, Y_{true}, \Theta\right)\right), \tag{4}$$

where $N(\cdot)$ is the normalized function. Note that in order to prevent the ultimate word embeddings from being too far apart from the original word embedding space, we map the perturbed word embedding to a valid word embedding in each iteration. This also ensures that the semantics of the words would not change significantly, thus leading to a higher semantic similarity of the adversarial examples.

Specifically, for word w_i and the perturbed word embedding $E_{w_i^{adv}}$, we first use cosine similarity and POS filtering to select the synonyms of w_i in the vocabulary $\mathbb{V}$ to form a candidate set $\mathbb{V}_{w_i}$. The maximum size of the set is $candidate_k$. Then we calculate the cosine similarity between the word embedding of the word in the candidate set and $E_{w_i^{adv}}$, and those below the threshold θ_{CAN} will be discarded. The word with the largest score w.r.t the objective function will be used as the final candidate word w^*.

3.1.2. Beam Search with Multiple Random Perturbing Positions

In fact, the gradient direction of the surrogate model is not absolutely accurate. To obtain a higher attack success rate and semantic similarity, we design an improved beam search in our algorithm. Specifically, for each sentence, we randomly select N words and sort their candidate words in descending order, where N is the number of perturbing positions. For each word, we select the top k candidate words with the largest similarity to substitute, get $k \cdot N$ sentences, and keep b sentences with the highest score w.r.t. the objective function, where b is the beam width. These random perturbing positions not only increase the search range, but also alleviate the error caused by the word importance ranking method which is widely used in previous works. Further discussion can be found in Section 6.

For the adversarial examples we finally obtain, we also check their semantic similarity. Using Universal Sentence Encoder (USE) [28], we encode the adversarial example and the original example into high-dimensional vectors and calculate the cosine similarity between them. Only the ones greater than the threshold θ_{USE} will be retained. Otherwise, the attack will be considered a failure. If multiple adversarial examples meet the conditions, then we choose the one with the highest semantic similarity.

3.1.3. Objective Function in Beam Search

In previous work, the attack effect on model precision is commonly used as the objective function h for adversarial attacks. It can be formally defined as

$$h(X, X_{adv}) = F_{Y_{true}}(X) - F_{Y_{true}}(X_{adv}). \tag{5}$$

Furthermore, we combine semantic similarity into the objective function. A BERT-based encoder is used to encode original and adversarial examples. Semantic similarity is defined as

$$s(X, X_{adv}) = \cos(Encoder(X), Encoder(X_{adv})). \tag{6}$$

Finally, we normalize h as $1/(1+e^{-h})$ and obtain the following objective function

$$h_{sim} = \frac{1}{1+e^{-h}} \cdot s. \tag{7}$$

3.2. Surrogate Models

Usually, the training of a surrogate model requires feedback from the target model. However, in the black-box scenario, this would consume a large amount of computing resources, so it is difficult for the surrogate model to approximate the decision boundary of the target model accurately. Thus, we integrate multiple models of different popular structures, such as WordCNN [29], LSTM [30] and Transformers [1] and use the gradient of the integrated model for the perturbation direction. The rationale behind this comes from some recent studies which have shown that neural networks always do shortcut learning [31] on difficult examples [32,33], which allows some of the known vulnerabilities to be shared. The integrated model will highlight these vulnerabilities and provide a more effective perturbation direction accordingly. Specifically, for the weighted average of the gradient of all surrogate models, we have

$$G_{inte} = \Sigma_{i=1}^{k} \alpha_i G_{s_i}, s.t. \Sigma \alpha_i = 1, \tag{8}$$

where G_{inte} is the integrated gradient, G_{s_i} is the gradient of the i^{th} surrogate model, α_i is the weight of the i^{th} surrogate model which is greater than 0, and k is the number of surrogate models.

4. Experimental Setup

4.1. Datasets

IMDB [34] is a document-level sentiment analysis dataset containing 50,000 non-professional movie reviews. It is widely used in the field of text classification.

SST-2 [35] is a sentence-level sentiment analysis dataset. The movie reviews are given by professionals.

MultiNLI [36] is a challenging dataset in the natural language inference (NLI) field with high semantic complexity. It includes eval data matched and mismatched with training documents (MNLI m and MNLI mm).

4.2. Target Models

WordCNN consists of an embedding layer, a 1D-convolutional layer, a 1D-maxpooling layer, and a fully connected layer.

BiLSTM replaces the 1D-convolutional layer and 1D-maxpooling layer in the WordCNN model with a bidirectional LSTM layer.**Bidirectional Encoder Representations from Transformers** (BERT) [3] is a popular, large-scale pre-training model.

ESIM [37] is an excellent model for NLI tasks before the advent of BERT.

4.3. Attacking Baselines

PWWS [18] ranks the words using the Probability Weighted Word Saliency method and then greedily replaces the words.

TextFooler [19] performs word importance ranking and then greedily replaces words using synonyms (while restricting semantic similarity), POS tag, etc.

BAE-R [25] utilizes a masked language model (MLM) in the generation of adversarial texts to help generate grammatically correct candidate words.

To make the experiment fair, we modify the baselines so that the cosine similarity between the final adversarial example and the original example must be greater than the threshold θ_{USE}. We also report the results of the unmodified baselines.

4.4. Parameters of the Beam Search

To balance the efficiency and the attack success rate of our method, we explore the parameter settings of the beam search. By attacking 100 clean examples on the SST-2 dataset against the BERT-base model, we compare the performance of our method in different settings of b and k. (See Table 1).

As can be seen in Table 2, in general, with the increase of b and k, the attack performance is significantly improved. When k is large enough, such as 10, and as b increases from 1 to 5, the accuracy of the target model drops from 55% to 5%, which shows that the range of b has a crucial influence on the attack performance of our method. When b continues to increase, the accuracy of the target model does not decrease significantly, but the perturbation rate decreases slightly, and the semantic similarity improves.

We also try reducing the search range k of candidate words. We fix b to be 5. When k increases from 5 to 10, the decrease in robust accuracy is insignificant. The smaller the k, the closer the actual perturbation direction is to the gradient of the surrogate models, which shows that it is very accurate and effective to use the integrated gradient as the perturbation direction.

More importantly, as the search range expands, the semantic similarity of the adversarial examples gradually rises. This is not possible with previous methods based on the greedy replacement. Because for a specific replacement strategy and a clean sample, the adversarial sample crafted by these greedy methods is certain, while beam search performs a full exploration of the search space to find an adversarial example with higher semantic similarity.

Table 1
Summary of the datasets.

Dataset	Classes	Train	Test	Avg Len
IMDB	2	25,000	25,000	201
SST-2	2	6,920	1,821	17
MNLI	3	433,000	10,000	11

Table 2
Exploration of beam width b and number of candidate words k in the beam search on SST-2 dataset. BERT-base is used as the target model. The settings we adopt are underlined.

Parameters	Robust Acc % ↓	Pert % ↓	Sim ↑
$b = 1, k = 10$	55.0	16.04	0.853
$b = 2, k = 10$	23.0	14.86	0.887
$b = 3, k = 10$	16.0	14.62	0.889
$b = 4, k = 10$	9.0	13.86	0.899
$b = 5, k = 10$	<u>5</u>.0	<u>1</u>3.59	<u>0</u>.907
$b = 5, k = 9$	6.0	12.75	0.909
$b = 5, k = 8$	8.0	13.00	0.908
$b = 5, k = 7$	7.0	13.86	0.903
$b = 5, k = 6$	8.0	14.01	0.905
$b = 5, k = 5$	9.0	14.37	0.900
$b = 6, k = 10$	5.0	13.19	0.905
$b = 7, k = 10$	**4.0**	12.78	0.913
$b = 8, k = 10$	**4.0**	12.52	0.916
$b = 9, k = 10$	**4.0**	**11.97**	**0.923**
$b = 10, k = 10$	**4.0**	12.18	0.922

5. Results

We conduct sufficient experiments on two tasks: text classification and natural language inference. By using all the above attacking methods to generate 1000 adversarial examples respectively, we evaluate their performance based on the three automatic metrics of robust accuracy, perturbation rate, and semantic similarity.

5.1. Attacking Results

Table 3 summaries the overall attacking performance of our method. As can be seen from the table, our method reduces the accuracy of all models to less than 7% and keeps the perturbation rate very low. In particular, our method reduces the accuracy of the WordCNN model from 88.5% to 0.0% on the IMDB dataset with a perturbation rate of only 1.42%. The low perturbation rate makes the generated adversarial examples more difficult to detect. There is a higher perturbation rate on the SST-2 and the MNLI dataset, which is because the texts in these two datasets are much shorter than those in IMDB. The number of perturbed words on these two datasets is only two on average, which implies that our method maintains the semantics of the clean samples.

To our surprise, the BERT model, which is expected to be more robust, turns out to be extremely vulnerable to adversarial attacks, which implies that adversarial examples completely mislead the BERT model. Besides, on the SST-2 dataset, the robust accuracy of the BiLSTM model are higher than those of the BERT model. This not only reflects the robustness of LSTM but also reminds us that while large-scale pre-training models can improve performance, the issue of adversarial robustness has been largely ignored.

5.2. Benchmark Comparison

We make a detailed comparison between our proposed method and three black-box attacking methods based on the four metrics of robust accuracy, query number, perturbation rate, and semantic similarity. As shown in Table 4, we attack 1000 original examples each time. We also report the unmodified baselines' results in Table 5. Our method reduces the accuracy of all models to below 7%, which is the best among the four methods. Since our method uses beam search, slightly more queries are handed out to the target model. The difference in the perturbation rate is insignificant. Taking the length of the texts into account, the difference in the number of perturbed words does not exceed one. Moreover, our method always maintains a semantic similarity which is above 0.895, which outperforms the attacking baselines and affirms the superiority of our method.

Table 3
Overall performance of our method on different models and datasets, including the original accuracy on 1000 clean samples (“Orig Acc %”), the accuracy after adversarial attack (“Robust Acc %”), and the percentage of perturbed words w.r.t. the original sentence length (“Perturb %”).

Dataset	Target Model	Orig Acc %	Robust Acc %	Perturb %	Avg Len
IMDB	BERT-base	92.7	0.1	4.47	201
	WordCNN	88.5	0.0	1.42	
SST-2	BERT-base	90.1	3.5	13.66	17
	BiLSTM	77.0	4.6	10.71	
MNLI m	BERT-base	77.8	6.3	15.41	11
	ESIM	72.2	6.3	16.63	
MNLI mm	BERT-base	79.8	3.1	15.07	12
	ESIM	75.1	4.4	15.46	

Table 4
Comparison of our method and three state-of-the-art black-box attacking methods.

Dataset	Target Model	Method	Robust %	#Query	Perturb %	Sim
IMDB	BERT-base	PWWS	3.4	1359.56	5.20	0.965
		TextFooler	10.5	1174.60	7.22	0.970
		BAE-R	45.1	**559.69**	**3.43**	**0.989**
		Ours	**0.1**	1622.13	4.47	0.979
	WordCNN	PWWS	**0.0**	1179.81	1.83	0.989
		TextFooler	**0.0**	**548.20**	3.69	0.985
		BAE-R	7.9	1088.49	2.29	0.991
		Ours	**0.0**	873.86	**1.42**	**0.993**
SST-2	BERT-base	PWWS	24.2	135.43	15.07	0.881
		TextFooler	21.9	91.92	17.09	0.883
		BAE-R	44.1	**58.36**	15.47	0.904
		Ours	**3.5**	248.54	**13.66**	**0.908**
	BiLSTM	PWWS	13.6	129.85	13.55	0.892
		TextFooler	11.0	72.27	13.70	0.894
		BAE-R	20.2	**56.59**	12.80	0.845
		Ours	**4.6**	382.62	**10.71**	**0.895**
MNLI m	BERT-base	PWWS	28.4	168.20	16.56	0.861
		TextFooler	18.5	79.40	15.50	0.876
		BAE-R	41.2	**66.25**	17.03	0.896
		Ours	**6.3**	112.50	**15.41**	**0.904**
MNLI mm	BERT-base	PWWS	23.0	98.90	14.62	0.860
		TextFooler	16.6	88.00	14.70	0.880
		BAE-R	37.8	**64.70**	**14.16**	0.903
		Ours	**3.1**	123.72	15.07	**0.907**

Table 5
The attacking results of the unmodified baselines on the IMDB dataset.

Dataset	Target Model	Method	Robust %	#Query	Perturb %	Sim
IMDB	BERT-base	PWWS	2.9	1423.13	7.30	0.923
		TextFooler	8.5	1234.54	8.05	0.930
		BAE-R	22.9	621.13	4.45	0.944
	WordCNN	PWWS	0.0	1209.87	2.60	0.962
		TextFooler	0.0	601.10	4.01	0.950
		BAE-R	5.2	1271.14	3.70	0.971

Overall, our method has the best performance in attack success rate and semantic similarity, and the perturbation rate is equivalent to those state-of-the-art methods.

6. Discussion

6.1. Ablation Study

To illustrate the superiority of our proposed method, we conduct ablation experiments. The substitution strategy is divided into three parts: 1) replacement ordering; 2) candidates construction; and 3) search method to explore the impact of different combinations on attack effect. For replacement ordering, word importance ranking and random order are compared. For candidates construction, constraints like the number of synonyms and the semantic similarity threshold are set consistent.

In general, it can be seen in Table 6 that beam search has a higher attack success rate as well as a higher semantic similarity than greedy search. Our method of constructing candidates combined with transferability (“surrogate”) has a better attack performance. Though random replacement order is not intuitively a good strategy, it is surprising that the combination of random replacement order and beam search works well. It means that random strategies are more suitable for beam search, and word importance ranking is more suitable for greedy search.

Inspired by this, we add more random perturbing positions to increase the randomness. We expect this will further improve the attack performance. The results of the experiments confirm our hypothesis. A larger N (random perturbing positions) enables more vulnerable words to be discovered, which significantly

Table 6
Attack performance of different substitution strategy combinations on the SST-2 dataset against the BiLSTM model. Note that the multiple perturbing positions strategy is incompatible with greedy search; hence those combinations are not included. Abnormal data are underlined.

Replacement Ordering	Candidates $k = 10$	Search Method	Robust %	Perturb %	Sim
word importance ranking	embedding	greedy	42.0	18.17	0.864
word importance ranking	embedding	beam($b = 5$)	38.5	19.35	0.874
word importance ranking	wordnet	greedy	72.5	13.46	0.920
word importance ranking	wordnet	beam($b = 5$)	72.0	13.14	0.929
word importance ranking	embedding + surrogate	greedy	41.0	18.96	0.859
word importance ranking	embedding + surrogate	beam($b = 5$)	32.5	18.30	0.864
random($N = 1$)	embedding	greedy	46.5	18.28	0.863
random($N = 1$)	embedding	beam($b = 5$)	31.0	19.41	0.869
random($N = 1$)	wordnet	greedy	71.0	11.14	0.924
random($N = 1$)	wordnet	beam($b = 5$)	53.0	19.19	0.888
random($N = 1$)	embedding + surrogate	greedy	48.5	16.53	0.857
random($N = 1$)	embedding + surrogate	beam($b = 5$)	26.5	19.44	0.861
random($N = 5$)	embedding	beam($b = 5$)	16.5	17.24	0.890
random($N = 5$)	wordnet	beam($b = 5$)	19.5	16.23	**0.898**
random($N = 5$)	embedding + surrogate	beam($b = 5$)	**12.5**	**16.09**	**0.898**

improves the attack effect. Furthermore, we argue that the importance of words changes dynamically as the replacement continues, so the error of word importance ranking will increase correspondingly. However, beam search with a random strategy would adapt better to such a change, which gives rise to adversarial examples of better quality (semantic similarity).

It is worth mentioning that in the table, those attack performance numbers in red seem abnormal. These attack strategies have a very low attack success rate, which means that they fail to attack difficult examples which require more modification. Hence, it leads to a lower perturbation rate and a higher semantic similarity on easy examples.

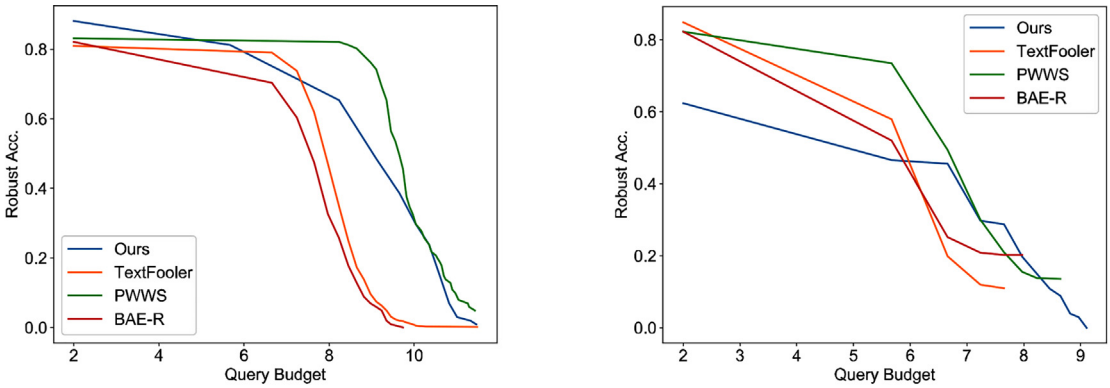
6.2. Evaluation of the trade-off

There is a trade-off between computational complexity and attack success rate. In general, beam search uses more queries to obtain a better attack performance. Thanks to our improvements to the beam search, our method can generate adversarial examples more efficiently, thereby making up for the search's original shortcoming in computational complexity.

In Fig. 1, we show the attack performance of our method and three advanced baselines under different query budgets. As can be seen from the figure, when the query budget is small, the attack performance gap between the four methods is insignificant. It is due to the application of multiple perturbing positions in beam search. As the query budget increases, our method can more fully explore the search space, and therefore yield a better attack performance.

6.3. Adversarial Training

It has been proved that adversarial training (AT) can improve the NLP model's generalization ability and adversarial robustness [38]. We attack 1000 clean examples on the MultiNLI and SST-2 training sets, and the generated adversarial examples are added to the original training sets. With the expanded data, we fine-tune BERT-base and BiLSTM and attack the models again to see whether it can obtain stronger adversarial robustness. As can be seen in Table 7, the adversarial attack's performance on the four metrics decreases, indicating that the model's adversarial robustness has been improved. In particular, the accuracy of the BiLSTM model increases from 4.6% to 22.0%, revealing the huge effect of AT. Because 1000 pieces of extra training data (adversarial examples) only account for a tiny portion (of about 0.2%) in the original MNLI training set, the BERT model's adversarial robustness is only slightly improved. (see Table 8).



(a) Robust accuracy of WordCNN on the IMDB dataset.

(b) Robust accuracy of BiLSTM on the SST-2 dataset.

Fig. 1. Robust accuracy of different victim models with different query budgets. The x-coordinate is in log-2 scale.

Table 7
Comparison of robust accuracy (“Acc %”), perturbation rate (“Pert %”), and semantic similarity (“Sim”) of original training and adversarial training (“+ AT”) models on MultiNLI and SST-2 dataset.

Dataset	Target	Acc %	Pert %	Sim
MNLI	BERT	3.1	16.07	0.907
mm	+ AT	3.9	16.38	0.904
SST-2	LSTM	4.6	17.02	0.895
	+ AT	22.0	25.90	0.863

Table 8
Generated adversarial examples by our method and other three baselines. The attack was failed if their label was the same with the original text.

IMDB	Label	Text
Original	Negative	This movie was soooooooo sloooow and everything in it was bland, the acting, the plot, etc. It was such a disappointment since the description looked so good. Do not be fooled. This movie is not worth the time it takes to watch it.
PWWS	Negative	This motion was soooooooo sloooow and everything in it was flat, the playacting, the game, etc. It was such a disappointment since the description looked thusly proficient. Arrange not be fooled. This motion is not worth the meter it takes to keep it.
TextFooler	Positive	This movie was soooooooo sloooow and everything in it was bland, the acting, the plot, etc. It was such a disappointment since the description looked so good. Do not be puzzled. This movie is not worth the time it takes to listen it.
BAE-R	Negative	This movie is so sloooow and everything in it was bland, the acting, the plot, etc. It was such a disappointment since the description looked so good. Do not be fooled. This film is not worth the time it takes to watch it.
Ours	Positive	This movie was soooooooo sloooow and everything in it was bland, the acting, the plot, etc. It was such a disappointment but the description looked so good. Do not be fooled. This movie is not worth the time it takes to watch it.

6.4. Case Study

To better present the naturalness and semantic similarity of our generated texts, we show some adversarial examples generated by our proposed method and other baselines.

7. Conclusion and Future Work

In this paper, we utilize an improved beam search and transferability from surrogate models to perform textual adversarial attacks under the black-box setting. Experimental results illustrate that our proposed method achieves a high attack success rate while keeping intact the original semantics. However, some candidates generated from the word embeddings are not comprehensive enough, and were omitted. Therefore utilizing large scale language models to generate more semantic-preserving candidates can be promising in the future.

CRediT authorship contribution statement

Bin Zhu: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - original draft. **Zhaoquan Gu:** Data curation, Writing - original draft, Funding acquisition. **Yaguan Qian:** Formal analysis, Investigation. **Francis Lau:** Writing - review & editing. **Zhihong Tian:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank the anonymous reviewers for their very helpful comments which helped improve the presentation of this paper. This work is supported in part by the National Key R&D Program of China (No. 2019YFB1706003), the Major Key Project of PCL (No. PCL2022A03), the Key Program of Zhejiang Provincial Natural Science Foundation of China (No. LZ22F020007), the Natural Science Foundation of China (No. 61902082), the Guangdong Province Key R&D Program of China (No. 2019B010136003), and the Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019).

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, CoRR abs/1706.03762. arXiv:1706.03762.
[2] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237. doi:10.18653/v1/N18-1202. https://www.aclweb.org/anthology/N18-1202.
[3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423. https://www.aclweb.org/anthology/N19-1423.
[4] H. Liu, Q. Yin, W.Y. Wang, Towards explainable NLP: A generative explanation framework for text classification, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5570–5581, https://doi.org/10.18653/v1/P19-1560.
[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, I. Erhan, Dumitru; Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199.
[6] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572.
[7] J. Gao, J. Lanchantin, M.L. Soffa, Y. Qi, Black-box generation of adversarial text sequences to evade deep learning classifiers, arXiv preprint arXiv:1801.04354.
[8] N. Papernot, P.D. McDaniel, I.J. Goodfellow, S. Jha, Z.B. Celik, A. Swami, Practical black-box attacks against machine learning, in: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2–6, 2017, ACM, 2017, pp. 506–519.
[9] Y. Liu, X. Chen, C. Liu, D. Song, Delving into transferable adversarial examples and black-box attacks, CoRR abs/1611.02770. arXiv:1611.02770.
[10] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial examples in the physical world, CoRR abs/1607.02533. arXiv:1607.02533.
[11] J. Li, W. Monroe, D. Jurafsky, Understanding neural networks through representation erasure, CoRR abs/1612.08220. arXiv:1612.08220.
[12] N. Papernot, P.D. McDaniel, A. Swami, R.E. Harang, Crafting adversarial input sequences for recurrent neural networks, CoRR abs/1604.08275. arXiv:1604.08275.
[13] S. Samanta, S. Mehta, Towards crafting text adversarial samples, CoRR abs/1707.02812. arXiv:1707.02812.
[14] J. Li, S. Ji, T. Du, B. Li, T. Wang, Textbugger: Generating adversarial text against real-world applications, CoRR abs/1812.05271. arXiv:1812.05271.
[15] Z. Gong, W. Wang, B. Li, D. Song, W.-S. Ku, Adversarial texts with gradient methods, CoRR abs/1801.07175. arXiv:1801.07175.
[16] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, K.-W. Chang, Generating natural language adversarial examples, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 2890–2896, https://doi.org/10.18653/v1/D18-1316.

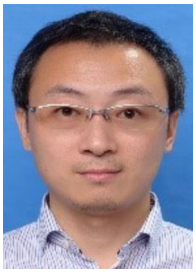
[17] M. Sato, J. Suzuki, H. Shindo, Y. Matsumoto, Interpretable adversarial perturbation in input embedding space for text, CoRR abs/1805.02917. arXiv:1805.02917.
[18] S. Ren, Y. Deng, K. He, W. Che, Generating natural language adversarial examples through probability weighted word saliency, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 1085–1097, https://doi.org/10.18653/v1/P19-1103.
[19] D. Jin, Z. Jin, J. Tianyi Zhou, P. Szolovits, Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment, arXiv e-prints (2019) arXiv:1907.11932 arXiv:1907.11932..
[20] Y. Zang, F. Qi, C. Yang, Z. Liu, M. Zhang, Q. Liu, M. Sun, Word-level textual adversarial attacking as combinatorial optimization, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 6066–6080, https://doi.org/10.18653/v1/2020.acl-main.540.
[21] L. Li, R. Ma, Q. Guo, X. Xue, X. Qiu, BERT-ATTACK Adversarial attack against BERT using BERT, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2020, pp. 6193–6202, https://doi.org/10.18653/v1/2020.emnlp-main.500.
[22] Y. Belinkov, Y. Bisk, Synthetic and natural noise both break neural machine translation, CoRR abs/1711.02173. arXiv:1711.02173.
[23] X. Wang, H. Jin, K. He, Natural language adversarial attacks and defenses in word level, CoRR abs/1909.06723. arXiv:1909.06723.
[24] R. Jia, A. Raghunathan, K. Göksel, P. Liang, Certified robustness to adversarial word substitutions, 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 4129–4142, https://doi.org/10.18653/v1/D19-1423.
[25] S. Garg, G. Ramakrishnan, BAE: BERT-based adversarial examples for text classification, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6174–6181. doi:10.18653/v1/2020.emnlp-main.498. https://www.aclweb.org/anthology/2020.emnlp-main.498.
[26] W. Hu, Y. Tan, Black-box attacks against RNN based malware detection algorithms, CoRR abs/1705.08131. arXiv:1705.08131.
[27] T. Zhao, Z. Ge, H. Hu, D. Shi, Generating natural language adversarial examples through an improved beam search algorithm, CoRR abs/2110.08036. arXiv:2110.08036. https://arxiv.org/abs/2110.08036.
[28] D. Cer, Y. Yang, S.-Y. Kong, N. Hua, N. Limtiaco, R. St, N. John, M. Constant, S. Guajardo-Cespedes, C. Yuan, B. Tar, R. Kurzweil Strope, Universal sentence encoder for English, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 169–174, https://doi.org/10.18653/v1/D18-2029.
[29] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751. doi:10.3115/v1/D14-1181. https://www.aclweb.org/anthology/D14-1181.
[30] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780. arXiv:https://doi.org/10.1162/neco.1997.9.8.1735.
[31] R. Geirhos, J. Jacobsen, C. Michaelis, R.S. Zemel, W. Brendel, M. Bethge, F.A. Wichmann, Shortcut learning in deep neural networks, CoRR abs/2004.07780. arXiv:2004.07780.
[32] C. Zhao, P.T. Fletcher, M. Yu, Y. Peng, G. Zhang, C. Shen, The adversarial attack and detection under the fisher information metric, CoRR abs/1810.03806. arXiv:1810.03806.
[33] D. Datta, S. Kumar, L.E. Barnes, T. Fletcher, Geometry matters: Exploring language examples at the decision boundary, CoRR abs/2010.07212. arXiv:2010.07212.
[34] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 142–150. https://www.aclweb.org/anthology/P11-1015.
[35] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 1631–1642.
[36] A. Williams, N. Nangia, S.R. Bowman, A broad-coverage challenge corpus for sentence understanding through inference, in: Proceedings of NAACL-HLT, 2018, pp. 1112–1122.
[37] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, D. Inkpen, Enhanced LSTM for natural language inference, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1657–1668, https://doi.org/10.18653/v1/P17-1152.
[38] T. Miyato, A.M. Dai, I.J. Goodfellow, Virtual adversarial training for semi-supervised text classification, CoRR abs/1605.07725. arXiv:1605.07725.

Bin Zhu received his bachelor degree in Software Engineering from Southwest Jiaotong University (2019). He is currently earning his master's degree in the Cyberspace Institute of Advanced Technology (CIAT), Guangzhou University, China. His research includes Natural Language Processing and Adversarial Robustness.

Zhaoquan Gu received his bachelor degree in Computer Science from Tsinghua University (2011) and PhD degree in Computer Science from Tsinghua University (2015). He is currently a Professor in the Cyberspace Institute of Advanced Technology (CIAT), Guangzhou University, China. His research includes wireless networks, distributed computing, and big data analysis.

Yaguan Qian received the B.S. degree in computer science from Tianjin University, in 1999, and the M.S. and Ph.D. degrees in computer science from Zhejiang University, in 2005 and 2014, respectively. He is an Associate Professor with the School of Big Data Science, Zhejiang University of Science and Technology. In his research areas, he has published more than 30 articles in international journals or conferences. His research interests mainly include machine learning, big-data analysis, pattern recognition, and machine vision. He is a member of the China Computer Federation (CCF), the Chinese Association for Artificial Intelligence (CAAI), and The Chinese Association of Automation (CAA).

Francis C.M. Lau received his PhD in computer science from the University of Waterloo in 1986. He has been a faculty member of the Department of Computer Science, The University of Hong Kong since 1987, where he served as the department chair from 2000 to 2005. He is now Associate Dean of Faculty of Engineering, the University of Hong Kong. He was a honorary chair professor in the Institute of Theoretical Computer Science of Tsinghua University from 2007 to 2010. His research interests include computer systems and networking, algorithms, HCI, and application of IT to arts. He is the editor-in-chief of the Journal of Interconnection Networks.



Zhihong Tian (Member, IEEE) is currently a Professor and the Dean of the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangdong Province, China. He is a Distinguished Professor with Guangdong Province Universities and Colleges Pearl River Scholar. He is also a part-time Professor with Carlton University, Ottawa, Canada. Previously, he served in different academic and administrative positions with the Harbin Institute of Technology. His research has been supported in part by the National Natural Science Foundation of China, National Key research and Development Plan of China, National High-tech Research and Development Program of China (863 Program), and National Basic Research Program of China (973 Program). He has authored over 200 journal and conference papers in these areas. His research interests include computer networks and cyberspace security. He also served as a member, a chair, and a general chair of a number of international conferences. He is a Senior Member of the China Computer Federation.