ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9747475

BLOCK-SPARSE ADVERSARIAL ATTACK TO FOOL TRANSFORMER-BASED TEXT CLASSIFIERS

Sahar Sadrizadeh[*], *Ljiljana Dolamic*[†], *and Pascal Frossard*[*]

[*]École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[†]Armasuisse S+T, Thun, Switzerland

ABSTRACT

Recently, it has been shown that, in spite of the significant performance of deep neural networks in different fields, those are vulnerable to adversarial examples. In this paper, we propose a gradient-based adversarial attack against transformer-based text classifiers. The adversarial perturbation in our method is imposed to be block-sparse so that the resultant adversarial example differs from the original sentence in only a few words. Due to the discrete nature of textual data, we perform gradient projection to find the minimizer of our proposed optimization problem. Experimental results demonstrate that, while our adversarial attack maintains the semantics of the sentence, it can reduce the accuracy of GPT-2 to less than 5% on different datasets (AG News, MNLI, and Yelp Reviews). Furthermore, the block-sparsity constraint of the proposed optimization problem results in small perturbations in the adversarial example. [1]

Index Terms— Adversarial attack, block sparse, deep neural network, natural language processing, text classification.

1. INTRODUCTION

In recent years, with the emerging high computational devices, Deep Neural Networks (DNNs) have attracted tremendous attention in many different fields such as computer vision [1] and Natural Language Processing (NLP) [2] due to their great performance. However, it has been shown that these models are highly vulnerable to perturbation of input samples, in particular to adversarial examples [3]. These examples, which are generated by making small or often imperceptible changes to the original input, can mislead the learning model to classify the adversarial example into a wrong predetermined target class (targeted attack) or to a different class than the true one (untargeted attack). Recently, many methods have been proposed to generate adversarial examples in image data to make the systems fail [4, 5], but these methods cannot be directly extended to NLP models, due to both the different nature of the data representation and the difficulty of characterizing imperceptible changes in text.

In visual applications, the main methods for generating adversarial examples are based on optimization and gradient descent. However, this is not readily extendable to textual data due to its specific nature. Therefore, there exists only a few white-box attacks, which have access to the parameters and gradients of the system, against NLP models. Although it is not possible to calculate the gradients in the discrete space of textual data, it has been proposed to find the gradients in the embedding space, which is continuous. For example, Papernot et al. [6] replace random words in the input sentence with the nearest word in the embedding space whose difference with the original word is in the direction of the gradient. Sato et al. [7] extends Adv-Text [8] to generate adversarial perturbations by imposing the directions of the perturbations in the embedding space to align with meaningful embedding vectors. However, these methods may perturb many words in the sentence which makes the changes quite perceptible. On the other hand, Guo et al. [9], recently proposed Gradient-based Distributional Attack (GBDA) against text transformers. They consider a probability distribution over all the vocabulary for each word in the adversarial sentence. They optimize this continuous matrix of distribution to fool the target model. However, their proposed formulation is highly over-parameterized.

The second difficulty in dealing with textual data is the definition of imperceptibility of the adversarial attack. The ℓ_p-norm, which is common in images to measure the difference of adversarial example and the input, is not readily applicable in textual data. There are different definitions for imperceptibility of adversarial attack in the literature. Some approximate it by the number of edits in the original text [10, 11]. On the other hand, many attacks define imperceptibility as the semantic similarity between the adversarial example and the original input. They first select random words or find the most important words in the sentence based on different metrics such as the word saliency [12]. Afterwards, they replace the selected words with their synonyms [12, 13], other words with similar embedding vectors [14, 15], or words predicted by a masked language model [16]. However, most of these methods assume the black-box scenario and use heuristic strategies that result in sub-optimal performance.

In this paper, we propose a method based on gradient projection to generate token-level adversarial examples against transformer-based text classifiers. We assume the white-box scenario, which gives us access to the model parameters and

[1]The source code of our attack can be found at https://github.com/sssadrizadeh/transformer-text-classifier-attack

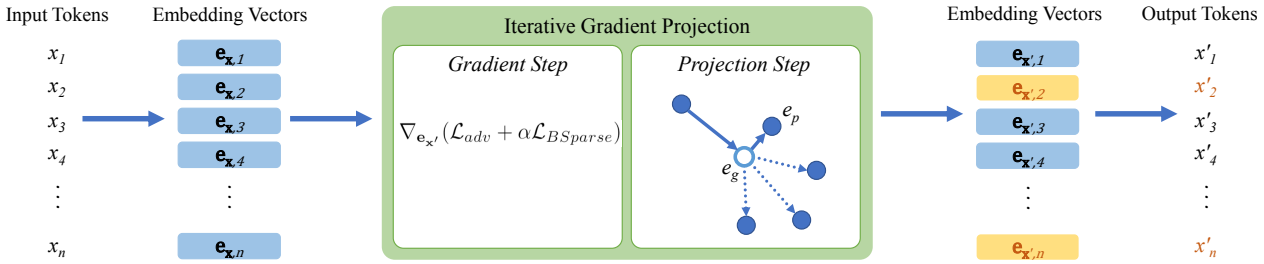

Fig. 1. Block diagram of the proposed method.

also their gradients. We consider perturbing the sequence of embedding vectors of the tokens in the input sentence. However, since we want only a few tokens to be changed, only a few blocks of the perturbation vector should be nonzero. Therefore, we add the block-sparsity constraint for the perturbation vector in the optimization problem. Moreover, we preserve the semantics of the sentence by projecting into the embedding vectors of the tokens which have the maximum cosine similarity with the embedding vectors of the corresponding original tokens. We evaluate our proposed attack against target transformer model with GPT-2 architecture [17] fine-tuned for different downstream NLP tasks such as natural language inference, sentiment analysis and news categorization. We compare our results with GDBA [9], a state-of-the-art white-box attack against text classifiers. To our knowledge, GDBA is the only white-box attack in the literature against transformer models. Experimental results indicate that the proposed adversarial attack achieves a competitive success rate in comparison to the GBDA method. Moreover, the projection to the closest token into the embedding space results in high semantic similarity between the adversarial example and the original sentence. Furthermore, our proposed block-sparsity constraint lead to small perturbations.

The rest of this paper is organized as follows. In Section 2, we formulate the problem of generating adversarial examples. Our attack algorithm is describe in Section 3. We evaluate our algorithm against different transformer models and discuss the experimental results in Section 4. Finally, the paper is concluded in Section 5.

2. PROBLEM FORMULATION

In this section, we present the formulation of generating adversarial example for textual data in untargeted attacks.

Consider $f : \mathcal{X} \to \mathcal{Y}$ to be the target text classifier model which correctly predicts the class of the input sentence $\mathbf{x} \in \mathcal{X}$ to be $y = f(\mathbf{x}) \in \mathcal{Y}$. Every sentence is considered to be tokenized to a sequence of tokens. We are looking for an adversarial example $\mathbf{x}'$, which differs from the input sentence $\mathbf{x}$ in only a few tokens and is semantically similar to it. However, the target model should classify $\mathbf{x}'$ wrongly, i.e., $f(\mathbf{x}') \neq y$.

Let $\mathbf{x} = x_1 x_2 ... x_n$ be the input sentence which is a sequence of n tokens of the vocabulary set $\mathcal{V}$. We assume that the adversarial example $\mathbf{x}' = x'_1 x'_2 ... x'_n$ is also a sequence of n tokens. The tokens of these sentences are in a discrete space. Therefore, each of these tokens is transformed to a continuous vector, called an embedding vector, as the input of the target transformer model [17]. Let emb(.) denote the embedding function that gets a token as the intput and transforms it to a continuous vector. Therefore, we can represent the sentence $\mathbf{x}$ in the embedding space as a sequence of embedding vectors $\mathbf{e_x} = [\text{emb}(x_1), \text{emb}(x_2), ..., \text{emb}(x_n)]$ by transforming each of its tokens by the function emb(.). Similarly, let $\mathbf{e_{x'}} = \mathbf{e_x} + \mathbf{r_x}$ represents the adversarial example as a sequence of embedding vectors. $\mathbf{r_x} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n]$ is the sequence of the perturbation vectors of each token.

Now, in order to fool the model with an untargeted attack, we can find an adversarial example by maximizing the loss function of the classifier, i.e., cross entropy. This is equivalent to finding the perturbed sample $\mathbf{e_{x'}}$ that minimizes the loss $\mathcal{L}_{Adv}$, which is defined as the negative of the cross entropy:

$$\mathcal{L}_{Adv} = -\mathcal{L}_f(\mathbf{e_{x'}}, y), \tag{1}$$

where $\mathcal{L}_f$ is the loss function of the model when the input is the adversarial example $\mathbf{e_{x'}}$ and the ground-truth class is y.

The above problem could lead to a large perturbation of the textual data. In order to constraint the changes to be small, we want to modify only a few tokens of the sentence. Therefore, only a few perturbation vectors (some blocks of $\mathbf{r_x}$) that correspond to the modified tokens are non-zero, while others are zero. In other words, the non-zero entries of the perturbation $\mathbf{r_x}$ occur in clusters, which means $\mathbf{r_x}$ should be block-sparse. To impose the block-sparsity of the perturbation, we can impose the sparsity on the norm of each block [18]. Hence, in the final optimization problem, we will minimize the ℓ_1 relaxation over all the ℓ_2 norms of perturbation blocks $\mathbf{r}_i$ to ensure the sparsity of non-zero blocks:

$$\mathcal{L}_{BSparse} = \sum_{i=1}^{n} \|\mathbf{r}_i\|_2. \tag{2}$$

Finally, we can reformulate the original optimisation problem of (1) by integrating the above block sparsity constraint. Therefore, our objective is to find the block-sparse perturbation that fool the target classifier by solving the following optimization problem:

$$\hat{\mathbf{e}_{\mathbf{x}'}} = \underset{\mathbf{e_{x'}} \in \mathcal{E}_{\mathcal{V}}}{\text{argmin}} \, \mathcal{L}_{Adv} + \alpha \mathcal{L}_{BSparse}, \tag{3}$$

where $\mathcal{E}_{\mathcal{V}}$ is the discrete subspace of every token of the vocabulary set $\mathcal{V}$ in the embedding space. Moreover, α is the

Algorithm 1 Block-Sparse Adversarial Attack

Input:
$f(.)$: Target classifier model, $\mathcal{V}$: Vocabulary set
$\mathbf{x}$: Tokenized input sentence, lr: Learning rate
A: Set of decreasing values for Hyper-parameter α to control the importance of the block-sparsity term
K: Maximum number of iterations
Output:
$\mathbf{x}'$: Generated adversarial example
procedure
initialization:
buffer $\leftarrow$ empty, $y \leftarrow f(\mathbf{x})$, $k \leftarrow 0$
$\forall i \in \{1, ...n\} \quad \mathbf{e}_{\mathbf{g},i} \leftarrow \text{emb}(x_i)$
for α in A **do**
while $f(\mathbf{e_p}) = y$ and $k \leq K$ **do**
$k \leftarrow k + 1$
Step 1: Gradient descent in the continuous embedding space:
$\mathbf{e_g} \leftarrow \mathbf{e_g} - lr.\nabla_{\mathbf{e}_{\mathbf{x}'}}(\mathcal{L}_{adv} + \alpha\mathcal{L}_{BSparse})$
Step 2: Projection to the discrete subspace $\mathcal{E}_\mathcal{V}$ and update if the sentence is new:
for $i \in \{1, ..., n\}$ **do**
$\mathbf{e}_{\mathbf{p},i} \leftarrow \underset{\mathbf{e}\in\mathcal{E}_\mathcal{V}}{\text{argmin}} \frac{\mathbf{e}^\top \mathbf{e}_{\mathbf{g},i}}{\|\mathbf{e}\|_2.\|\mathbf{e}_{\mathbf{g},i}\|_2}$
end for
if $\mathbf{e_p}$ not in **buffer then**
add $\mathbf{e_p}$ to **buffer**
$\mathbf{e_g} \leftarrow \mathbf{e_p}$
end if
end while
if $f(\mathbf{e_p}) \neq y$ **then**
break (adversarial example is found)
end if
end for
return $\mathbf{e}_{\mathbf{x}'} \leftarrow \mathbf{e_p}$
end procedure

hyper-parameter that determines the relative importance of the block-sparsity term.

3. PROPOSED METHOD

In this section, we explain our algorithm to find the solution of the proposed optimization problem (3). The block diagram of our method can be found in Figure 1. As depicted in this figure, we first transform each token of the input sentence to a continuous embedding vector and then we use gradient projection to solve the optimization problem (3).

Since we are dealing with textual data, (3) is a discrete optimization problem. In other words, the tokens of the resultant adversarial example should be in the vocabulary set $\mathcal{V}$; hence $\mathbf{e}_{\mathbf{x}'}$ should be in the discrete subspace $\mathcal{E}_\mathcal{V}$. First, we consider $\mathbf{e}_{\mathbf{x}'}$ to be in the embedding space $\mathcal{E}$ (and not necessarily in $\mathcal{E}_\mathcal{V}$). Thus, we can perform gradient descent to solve the optimization problem (3). In each iteration of our algorithm, we first update the embedding vectors of all the tokens of the adversarial example in the continuous space $\mathcal{E}$. Let $\mathbf{e}_{\mathbf{g},i}$ denote this updated vector in the continuous space corresponding to the i-th token. Afterwards, we project the updated embedding vectors $\mathbf{e}_{\mathbf{g},i}$, which may not necessarily correspond to a token in the vocabulary $\mathcal{V}$, to the embedding vectors of the closest meaningful tokens. We use cosine similarity metric to find the closest embedding vectors in $\mathcal{E}_\mathcal{V}$ and apply the projection for each token independently:

$$\forall i \in \{1, ..., n\}: \quad \mathbf{e}_{\mathbf{p},i} = \underset{\mathbf{e}\in\mathcal{E}_\mathcal{V}}{\text{argmin}} \frac{\mathbf{e}^\top \mathbf{e}_{\mathbf{g},i}}{\|\mathbf{e}\|_2.\|\mathbf{e}_{\mathbf{g},i}\|_2}. \quad (4)$$

Furthermore, since we are dealing with discrete data, it is possible that through iterations we come across a previously computed embedding vector after the projection. Moreover, if the perturbation vector is too small, the updated vectors will be projected to the previous sentence. In these cases, the algorithm will be stuck in a loop as the computed gradients will stay the same. To prevent such undesirable scenarios, we update the embedding vectors by the projection step only when the projected sentence has not been generated before. To this end, we save all the updated sentences in a buffer, and update the embedding vectors by the projected ones only if the output of the projection step is not in the buffer. These steps are performed iteratively until the target model is fooled or a maximum number of iterations is reached (the algorithm fails to find an adversarial example in this case).

As another consideration, we do not fix the value of α in (3), which determines the importance of imperceptibility. For higher values of α, the algorithm tries to find adversarial examples with smaller token error rate that are more similar to the original sentence. However, by increasing α, the success rate of finding an adversarial example decreases. Therefore, we will consider a large value for this hyper-parameter at first. If the algorithm fails to find an adversarial example, we will decrease the value of α. Algorithm 1 presents the pseudo-code of the proposed method for finding the minimizer of the discrete optimization problem (3).

4. EXPERIMENTAL RESULT

In this section, we evaluate our proposed adversarial attack in different text classification tasks such as natural language inference, sentiment classification, and news categorization.

Datasets. We evaluate our proposed method on the test set of three datasets: MNLI[2] [19] (natural language inference), AG News [20] (news categorization), and Yelp Reviews [20] (sentiment classification). Some statistics of these datasets can be found in Table 2.

Model. For the target model, we fine-tuned a pre-trained transformer model with GPT-2 architecture on all the mentioned datasets.

Baseline. We compare the result of our method with that of GDBA [9]. To our knowledge, GDBA is the only white-box attack in the literature against transformer models.

Hyper-parameters. We use Adam optimizer to find the minimizer of the proposed optimization problem with learning rate $\{0.15, 0.3\}$. Moreover, the coefficient α in (3) is in the set $\{10, 8, 5, 2\}$ divided by the length of the sentence. For

[2]We evaluate our method on the matched validation set of MNLI dataset.

Table 1. Examples of successful adversarial examples on different datasets.

Dataset	Sentence	Prediction	Text
MNLI	Original	Neutral (97.26%)	Premise: In the summer, the Sultan's Pool, a vast outdoor amphitheatre, stages rock concerts or other big-name events. Hypothesis: **Most** rock concerts take place in the Sultan's Pool amphitheatre.
	Adversarial	Entailment (99.19%)	Premise: In the summer, the Sultan's Pool, a vast outdoor amphitheatre, stages rock concerts or other big-name events. Hypothesis: **Many** rock concerts take place in the Sultan's Pool amphitheatre.
AG News	Original	Sci/Tech (99.39%)	Motorola and HP in **Linux** tie-up Motorola plans to sell mobile phone network equipment that uses **Linux**-based code, a step forward in network gear makers #39; efforts to rally around a standard.
	Adversarial	Business (83.56%)	Motorola and HP in **PC** tie-up Motorola plans to sell mobile phone network equipment that uses **PC**-based code. a step forward in network gear makers #39; efforts to rally around a standard.
Yelp	Original	Negative (99.90%)	This place holds a nostalgic appeal for people born and raised in Pittsburgh who grew up eating here. If that experience is what your looking for, please visit. If you're looking for a tasty meal, go somewhere else. 5 stars for history, **0** for food quality and flavor.
	Adversarial	Positive (96.54%)	This place holds a nostalgic appeal for people born and raised in Pittsburgh who grew up eating here. If that experience is what your looking for, please visit. If you're looking for a tasty meal, go somewhere else. 5 stars for history,**1** for food quality and flavor.

Table 2. Some statistics of the evaluation datasets. Last column (Clean Acc.(%)) is the accuracy of the fine-tuned GPT-2.

Dataset	Avg. Length	#Classes	Test Set	Clean Acc.
Ag News	43	4	7600	94.8
MNLI	11	3	9815	81.7
Yelp Reviews	157	2	38000	97.8

larger values of learning rate and smaller values of α, the attack is more aggressive and more words of the sentence are modified. Therefore we will change them in the mentioned sets if only the attack fails.

Evaluation. To evaluate our adversarial attack, we solve (3) by algorithm 1 and attack the target model with GPT-2 architecture which is fine-tuned over one of the aforementioned datasets. The MNLI dataset consists of sentence pairs, premise and hypothesis, and we attack them separately. Table 3 shows the result of our method in comparison with GDBA in terms of accuracy of the target model after the adversarial attack and semantic similarity. Semantic similarity between the adversarial example and the original sentence is computed by Universal Sentence Encoders [21] as is common in the literature. It is worth mentioning that we consider that our attack has failed if the semantic similarity is less than a threshold. The results show that our attack successfully drop the accuracy of the target model to less than 5% for all of the datasets while the semantic similarity is preserved (more than 0.8 in all cases). Compared to GBDA, success rate of our attack is superior in all cases, except for the MNLI dataset, in which our method achieves competitive results. Table 1 also shows some adversarial examples against different datasets generated by our method.

We investigate the effect of the hyper-parameter α in the optimization problem (3) on the performance of our method on AG News dataset. Figure 2 depicts the effect of this hyper-parameter on the accuracy of the target model, semantic similarity, and token error rate. By increasing α, success rate of our attack decreases while semantic similarity increases and the token error rate decreases. It is worth mentioning that we fix the learning at 0.15 for this experiment. Therefore, the accuracy is lower than the one reported in Table 3.

Table 3. Performance of white-box attack against the fine-tuned GPT-2 in terms of after attack accuracy (Adv. Acc.(%)) and semantic similarity (Sim.). For the MNLI dataset, the results of attacking the hypothesis sentences are in brackets.

Method	Ag News		MNLI		Yelp Reviews	
	Adv. Acc.	Sim.	Adv. Acc.	Sim.	Adv. Acc.	Sim.
Proposed	0.4	0.87	3.0 (1.3)	0.85 (0.82)	1.8	0.87
GBDA	6.6	0.90	2.8 (11.0)	0.82 (0.88)	2.9	0.94

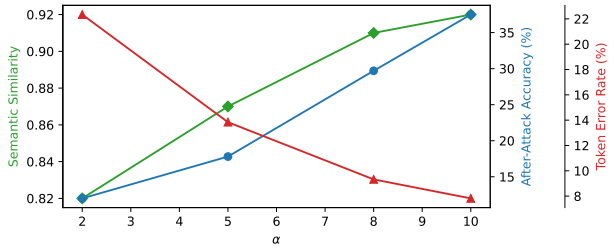

Fig. 2. Effect of the hyper-parameter α on the performance.

5. CONCLUSION

In this paper, we proposed a new white-box attack based on gradient projection against text classifiers. We proposed an optimization problem with a block-sparsity constraint to ensure that only a few words of the sentence are modified. Experimental results show that our attack is highly effective on fooling text classifiers in different tasks and it preserves the semantics of the sentence. In all tasks, the accuracy of the target model drops to less than 5% and the semantic similarity is more than 80%. We also compared our attack with GDBA, the only white-box attack against transformers. The success rate of our attack is superior to GDBA in all cases except for the MNLI dataset, in which our method achieves comparable results to GDBA.

6. REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[3] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[4] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[5] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations, ICLR 2018*, 2018.

[6] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang, "Crafting adversarial input sequences for recurrent neural networks," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 49–54.

[7] Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto, "Interpretable adversarial perturbation in input embedding space for text," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 4323–4330.

[8] Takeru Miyato, Andrew M Dai, and Ian Goodfellow, "Adversarial training methods for semi-supervised text classification," *arXiv preprint arXiv:1605.07725*, 2016.

[9] Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela, "Gradient-based adversarial attacks against text transformers," *arXiv preprint arXiv:2104.13733*, 2021.

[10] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou, "Hotflip: White-box adversarial examples for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 31–36.

[11] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 50–56.

[12] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che, "Generating natural language adversarial examples through probability weighted word saliency," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 1085–1097.

[13] Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun, "Word-level textual adversarial attacking as combinatorial optimization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6066–6080.

[14] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang, "Generating natural language adversarial examples," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2890–2896.

[15] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits, "Is bert really robust? a strong baseline for natural language attack on text classification and entailment," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, pp. 8018–8025.

[16] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu, "Bert-attack: Adversarial attack against bert using bert," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6193–6202.

[17] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, pp. 9, 2019.

[18] Ehsan Elhamifar and René Vidal, "Block-sparse recovery via convex optimization," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4094–4107, 2012.

[19] Adina Williams, Nikita Nangia, and Samuel Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1112–1122.

[20] Xiang Zhang, Junbo Zhao, and Yann LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, pp. 649–657, 2015.

[21] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al., "Universal sentence encoder," *arXiv preprint arXiv:1803.11175*, 2018.