Bridge the Gap Between CV and NLP! A Gradient-based Textual Adversarial Attack Framework

Lifan Yuan[1,2*†], **Yichi Zhang**[1,2*†], **Yangyi Chen**[2], **Wei Wei**[1,2‡]
[1]Cognitive Computing and Intelligent Information Processing Laboratory,
School of Computer Science and Technology Huazhong University of Science and Technology
[2]Huazhong University of Science and Technology
{lievanyuan173, phantivia, yangyichen6666}@gmail.com
weiw@hust.edu.cn

Abstract

Despite great success on many machine learning tasks, deep neural networks are still vulnerable to adversarial samples. While gradient-based adversarial attack methods are well-explored in the field of computer vision, it is impractical to directly apply them in natural language processing due to the discrete nature of text. To bridge this gap, we propose a general framework to adapt existing gradient-based method to craft textual adversarial samples. In this framework, gradient-based continuous perturbations are added to the embedding layer and are amplified in the forward propagation process. Then the final perturbed latent representations are decoded with a mask language model head to obtain potential adversarial samples. In this paper, we instantiate our framework with **T**extual **P**rojected **G**radient **D**escent (**TPGD**). We conduct comprehensive experiments to evaluate our framework by performing transfer black-box attacks on BERT, RoBERTa and ALBERT on three benchmark datasets. Experimental results demonstrate our method achieves an overall better performance and produces more fluent and grammatical adversarial samples compared to strong baseline methods. All the code and data will be made public.

1 Introduction

Deep learning has achieved great success in various domains, such as computer vision (CV) (He et al., 2016;Chi et al., 2019), natural language processing (NLP) (Vaswani et al., 2017; Devlin et al., 2019) and speech recognition (Chiu et al., 2018; Park et al., 2019). However, the powerful neural networks are still vulnerable to adversarial samples, crafted by adding small and human-imperceptible perturbations to the inputs. (Szegedy et al., 2014; Goodfellow et al., 2015).

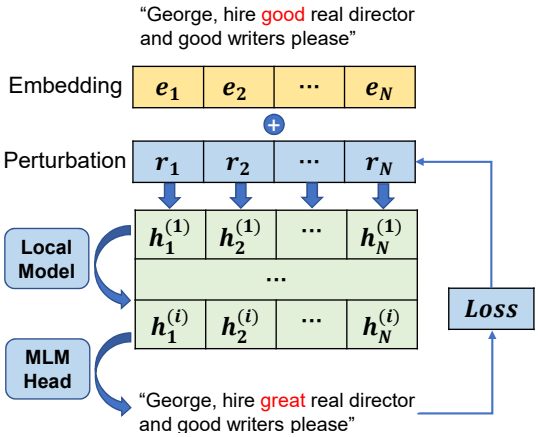

Figure 1: Overview of our framework.

In the field of CV, numerous adversarial attack methods have been proposed to evaluate the robustness of DNNs (Papernot et al., 2016a, Madry et al., 2019), and corresponding defence methods are also well-explored (Papernot et al., 2016c; Ross and Doshi-Velez, 2018). Adversarial attacks on images are defined as an optimization problem of maximizing the loss function of model on specific samples, which can be approximated by the gradient ascent algorithms.

However, textual adversarial attack is more challenging due to the discrete and non-differentiable nature of the text space. And the methods that directly employ the gradients to craft adversarial samples is not applicable in NLP. Current practices of textual adversarial attacks that employ first-order approximation to find substitute words are less effective for one-off searching and can violate the local linearization assumption (Cheng et al., 2019; Behjati et al., 2019; Xu and Du, 2020).

To bridge this gap, we propose a general framework to adapt existing gradient-based method to NLP. We successfully obtain high-quality adversarial samples by conducting gradient-based search. Specifically, we employ the gradient of loss func-

*Work done during internship at CCIIP
†Equally contribution, alphabetical order by name
‡Corresponding author

tion with respect to the embeddings of input tokens to make perturbations on token embeddings rather than on the original text, thus transforming the problem of searching for adversarial samples from the discrete text space to the continuous differentiable embedding space. This provides the basis for applying gradient-based methods investigated in CV to craft textual adversarial samples. In this paper, we adapt PGD (Madry et al., 2019) algorithm within our framework to perform textual adversarial attacks, denoted as **TPGD**. We iteratively generate small perturbations following the gradient information and add them to the embedding layer. Then the forward propagation process will amplify the perturbations.

The natural question is how can we transform the perturbed token embeddings back to the discrete text? Although there exist some works exploring the feasibility of directly perturbing token embeddings (Miyato et al. (2017); Sato et al. (2018); Cheng et al. (2019); Behjati et al. (2019)), None of them succeed to map all points in the embedding space back to words, thus failing to generate human-imperceptible adversarial samples. However, recent work observes that the mask language model (MLM) head can reconstruct input sentences from their hidden states with high accuracy, even after models have been fine-tuned on specific tasks (Kao et al., 2021). Inspired by this, we employ a MLM head to decode the perturbed latent representations. With the extensive linguistic knowledge of MLM head, the coherence and grammaticality of adversarial samples can be guaranteed.

We conduct comprehensive experiments to evaluate the effectiveness of our method by performing transfer black-box adversarial attacks, where only the final decisions of victim models are accessible, against three victim models on three benchmark datasets. We use a local pre-trained language model to construct potential adversarial samples and then query the victim models for decisions. Experimental results demonstrate the effectiveness of our framework and TPGD algorithm. Specifically, TPGD significantly outperforms all baseline methods in terms of attack success rate, and produces more fluent and grammatical adversarial examples.

To summarize, the main contributions of this paper are as follows:

- We propose a general gradient-based textual adversarial attack framework based on continuous perturbations, bridging the gap between CV and NLP on the study of adversarial attacks. Common gradient-based attack methods in CV can be easily adapted to NLP within our framework.
- We propose a novel adversarial attack method called TPGD within our framework. We employ a local model to construct adversarial samples by iteratively perturbing its embedding layer through the gradient information, and accumulating these small perturbations to search for potential adversarial samples.
- We successfully handle the challenge of black-box attack where only the decisions of models are accessible, which is rarely investigated in NLP.

2 Related Work

2.1 Adversarial Attack in CV

In the field of computer vision, adding a small amount of perturbations to input images to mislead the classifier is possible (Szegedy et al., 2014). Based on this observation, various adversarial attack methods have been explored. FGSM (Goodfellow et al., 2015) crafts adversarial samples using the gradient of the model's loss function with respect to the input images. BIM (Kurakin et al., 2017) straightforwardly extends FGSM, iteratively applying adversarial noises multiple times with a smaller step size. MIM (Dong et al., 2018) exploits momentum when updating inputs, obtaining adversary samples with superior quality. PGD (Madry et al., 2019) employs uniform random noise as initialization. Both MIM and PGD are variants of BIM.

2.2 Adversarial Attack in NLP

Existing textual attack models can be roughly categorized into white-box and black-box attack models according to the accessibility to the victim models.

White-box attack models, also known as gradient-based attack models, assume that the attacker has full knowledge of the victim models, including model structures and all parameters. There are few application scenarios of white-box attack in real-world situations, so most white-box attack models are explored to reveal the weakness of victim models, including universal adversarial triggers (Wallace et al., 2019), fast gradient sign inspired methods (Ebrahimi et al., 2018; Papernot et al., 2016b).

Black-box attack models can be further divided into two different attack settings, i.e. score-based and decision-based. The first one assumes the attacker can obtain the decisions and corresponding confidence scores from victim models. Most research works on black-box attack focus on this setting, exploring different word substitution methods and search algorithms to reduce the victim models' confidence scores. The word substitution methods mainly focus on word embedding similarity (Jin et al., 2020), WordNet synonyms (Ren et al., 2019), HowNet synonyms (Zang et al., 2020), and Masked Language Model (Li et al., 2020a). The search algorithms involve greedy search algorithm (Ren et al., 2019; Jin et al., 2020), genetic algorithm (Alzantot et al., 2018), and particle swarm optimization (Zang et al., 2020). The other attack setting assumes the attacker can only obtain decisions from victim models, which is more challenging and less studied. Maheshwary et al. (2021) first substitutes some words in the input sentences to flip the labels and then conducts search based on genetic algorithm, expecting to find the most semantic preserved adversarial samples. Chen et al. (2021) propose a learnable attack agent trained by imitation learning to perform decision-based attack. There also exist some works exploring sentence-level transformation, including syntax (Iyyer et al., 2018) and text style (Qi et al., 2021), to launch attack.

Note that although we apply gradient based methods in CV, the gradients we employ to generate the perturbations are obtained from the local model rather than the victim model. We only have access to the decisions of victim models. Therefore, we consider our method as decision-based black-box attack.

3 Framework

In this section, we first give an overview of our framework. Then we detail the process of adding continuous perturbations and reconstructing text from perturbed latent representations.

3.1 Overview

We adopt an encoder-decoder framework to apply gradient-based attack methods in CV. Specifically, we use a local language model as the encoder, and its MLM head as the decoder. (shown in Figure 2). For each input text, we add gradient-based perturbations on its token embedding. The perturbations will be amplified in the forward propagation process. Then the final perturbed latent representations are decoded with a MLM head to obtain potential adversarial samples.

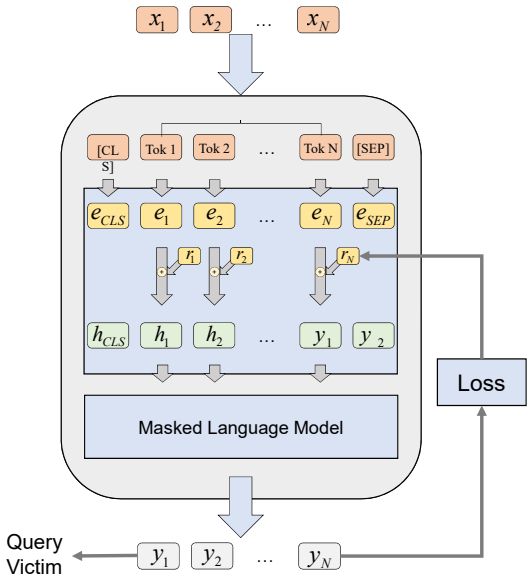

Figure 2: Overall demonstration of our framework. Continuous perturbations (r_i) are calculated as gradients of loss function with respect to token embeddings. The MLM head is employed to decode the final perturbed hidden states to obtain potential adversarial samples.

With this framework, we can adapt various gradient-based adversarial attack algorithms in CV to generate textual adversarial samples. In this paper we apply PGD (Madry et al., 2019) to generate gradient-based perturbations. Section 4 gives instantiation details.

3.2 Perturbation

We fine-tune a local BERT model and then adopt it as the encoder. For each input sample, we propagate it forward to calculate the current loss, and then perform backward propagation to obtain the gradients of the loss with respect to the token embeddings of input. The gradients are then employed to update the token embeddings (one step of perturbation). We solve the following optimization problem to find perturbations:

$$\delta = \underset{\delta:\|\delta\|_2 \leq \varepsilon}{\arg\max} \mathcal{L}\left(E + \delta, y; \theta\right), \qquad (1)$$

where δ is the perturbation, E stands for the representation embeddings of input tokens, y is the golden label, θ denotes current parameters of our

local model, and $\mathcal{L}(\cdot)$ is the loss function. Since it's hard to find the solution to equation (1) in a closed form, works in CV usually linearize the loss function with gradients information to approximate the perturbation δ (Goodfellow et al., 2015; Kurakin et al., 2017).

In NLP, most current gradient-based methods employ first-order approximation to obtain substitution words (Cheng et al., 2019; Behjati et al., 2019; Xu and Du, 2020). This one-off approach may result in large step size perturbations, violating the hypothesis of local linearization (See Figure 3). Different from previous works, we ensure that the local linearization assumption by directly adding small continuous perturbations to token embeddings, and iteratively accumulate the perturbations until the reconstructed text changes.

3.3 Reconstruction

After adding continuous perturbations to the token embeddings, we need to reconstruct the text from the hidden representations. Notice that the classification layer of MLM is observed to be able to reconstruct input sentences from hidden states in middle layers with high accuracy, even after model have been fine-tuned on specific tasks (Kao et al., 2021). Inspired by this, we adopt the MLM head as a decoder for: 1) MLM head is capable of interpreting any representation embeddings in the hidden space , which is crucial to search adversarial examples continuously; 2) MLM head has been fully trained during the pre-trained stage so it acquires linguistic knowledge together with the language model and can reconstruct sentences considering contextual information.

The discrepancy between previous one-off attacks and our iterative attack framework is demonstrated in Figure 3. We perturb token a to find its substitutes. After one step perturbation (denoted as $\vec{r_1}$), since $\cos(\overrightarrow{r_1}, \overrightarrow{ab}) < \cos(\overrightarrow{r_1}, \overrightarrow{ac})$, token b is thus considered as the substitute of token a in one-off attacks. However, in our approach, the perturbation does not cross the decoding boundary, i.e. the decoding results remains unchanged, hence the search continues. After iterative searches, the perturbation may cross the boundary and be decoded to the optimal solution token c, then we replace token a with token c and then query the victim model for its decision.

If the query fails, we start the next iteration of searching from the current perturbed token embedding, but not from the actual embedding of token c. This iterative attack is not based on substitution of existing actual tokens. Actually, it accumulates small perturbations along the gradient-based direction, which is more effective to find the optimal solution.

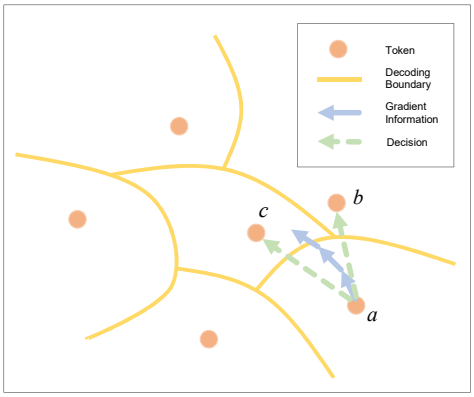

Figure 3: Search for the substitute token in the hidden space. One-off attacks are likely to select token b after one-step perturbation, while our iterative perturbations is more likely to find the optimal solution token c.

Moreover, since the input sentence is tokenized before being embedded to continuous representations, perturbation and reconstruction are operated on tokens which consist of affix, words and punctuation. Therefore, the hidden states are decoded to tokens rather than directly to words. Thus, we conduct adversarial attacks in a relatively smaller granularity, which has a great potential to better maintain the coherence and grammaticality of sentences.

4 Approach

We denote a pair of sample and label as $(x \in \mathcal{X}, y \in \mathcal{Y})$, the embedding of the token x as e, its hidden state as h, and the neural network as a function $f : \mathcal{X} \rightarrow \mathcal{Y}$. Further, f is divide into three parts f_0, f_1 and f_2, holding:

$$f(x) = f_2(f_1(f_0(x))), \tag{2}$$

where f_0 is the embedding layer, f_1 denotes the hidden layers that map embeddings to hidden states of a certain layer and f_2 denotes the rest of the neural network. Then the forward propagation process can be described as:

$$e = f_0(x), h = f_1(e), y = f_2(h) \tag{3}$$

4.1 Algorithm Flow

We instantiate our framework with PGD (Madry et al., 2019) algorithm, and name our textual adversarial attack model as **T**extual **PGD** (**TPGD**). To solve the above optimization problem (equation 1), we iteratively search perturbations and add them to the embedding layer, with the following formula:

$$\begin{aligned} g_{adv} &= \nabla_{\delta}\mathcal{L}\left(E, y; \theta\right) \\ \delta_{i+1} &= Proj\left(\delta_i + \alpha g_{adv}/\|g_{adv}\|_F\right), \end{aligned} \tag{4}$$

where g_{adv} is the gradient of loss with respect to δ, α is the perturbation step size, and i denotes the current iteration step. $Proj\left(\cdot\right)$ performs a re-initialization when δ reaches beyond the ϵ-neighborhood of the original embedding.

For each sample, we first map it to token embeddings, where continuous perturbations can be added to. After obtaining gradient of loss function with respect to the token embeddings in iteration $i+1$, perturbations δ_{i+1} are generated by equation (4) and then added to the the token embeddings. Then the perturbations are amplified through the forward propagation process. To craft textual adversarial samples, the perturbed hidden states will be decoded for reconstruction, which is denoted as:

$$adv_{i+1} = Dec(h_{i+1}), \tag{5}$$

where adv_{i+1} denotes the adversarial sample obtained in $i+1$ iteration. We query the victim model only when adv_{i+1} satisfying: 1) it varies from adv_0 to adv_i; 2) it is more similar to the original sentences, compared to previous potential adversarial samples. Here we employ the USE score to measure the similarity between sentences. If attack succeeds and $USE\left(adv_{i+1}, x\right) > T$, where T is a tunable threshold for USE score, then adv_{i+1} is considered as the adversarial sample of the original input.

4.2 Heuristic Methods

4.2.1 Random Mask Input Text for Diversity

To increase the diversity of adversarial samples, we randomly mask one token in each input sentence. Specifically, we tokenize x to a list of tokens, $x_{token} = [x_0, ..., x_{i-1}, x_i, x_{i+1}, ...]$, then randomly select an index i from the uniform distribution and replace x_i with token $[MASK]$. The MLM will predict the masked word given its context, thus diversifying the adversarial samples with semantically-consistent consideration. Then these processed discrete sentences are embedded into continuous token embeddings as mentioned. For each sample, the search has a maximum iteration to avoid the infinite loop.

4.2.2 Loss Function

Intuitively, the reconstruction accuracy of MLM head has a strong impact on the quality of adversarial sample. If the decoder fails to recover the original sentences even when no perturbations are added, its capacity to generate fluent adversarial samples from perturbed hidden states is limited. Thus, to avoid possible catastrophic drop on the quality of adversarial samples during the perturbation accumulation process, we need extra constraints on MLM head to ensure the reconstruction accuracy. Note that the MLM head has been pretrained to precisely fill the masked word, which fits well into our problem. Therefore, we also add a constraint that the added perturbations should minimize the loss of MLM while maximizing the loss of downstream tasks, so that the adversarial samples can fool the models with minimal perturbations. In all, the loss function consists of two parts:

$$\mathcal{L}\left(E, y; \theta\right) = \mathcal{L}_1\left(E, y; \theta\right) + \beta\mathcal{L}_2\left(E, y; \theta\right), \tag{6}$$

where $\mathcal{L}_1\left(E, y; \theta\right)$ is the original loss of the victim model on specific tasks, $\mathcal{L}_2\left(E, y; \theta\right)$ is the cross-entropy loss of MLM head , and β is a weighting constant. Note that we aim to reduce the decoding loss $\mathcal{L}_2$ while increasing $\mathcal{L}\left(E, y; \theta\right)$ along the gradient direction, so β should be negative. By taking two losses into account jointly, we adjust the perturbations search target on successfully fooling the victim models using fewer modifications.

4.2.3 Antonym Filter

Li et al. (2019) reports that semantically opposite words are quite close in their representation embeddings since antonyms usually appear in similar contexts. Therefore, we filter antonyms of original words using WordNet (Fellbaum, 2010) to prevent from crafting invalid adversarial samples.

5 Experiments

We conduct comprehensive experiments to evaluate our general framework and TPGD algorithm on the task of sentiment analysis, natural language inference, and news classification. We consider both automatic and human evaluations to analyze our method in terms of attack performance, semantic consistency, and grammaticality.

Dataset	#Class	Train	Test	Avg Len	BERT Acc	RoBERTa Acc	ALBERT Acc
SST-2	2	7K	1.8K	16.5	89.9	94.2	92.8
MNLI	3	433K	10K	31.7	82.8	83.6	82.3
AG's News	4	30K	1.9K	39.3	91.2	94.7	94.2

Table 1: Detail information of datasets and original accuracy of each victim models.

Dataset	Model	BERT				RoBERTa				ALBERT			
		ASR%	USE	ΔI	ΔPPL	ASR%	USE	ΔI	ΔPPL	ASR%	USE	ΔI	ΔPPL
SST-2	Textfooler	66.56	0.78	4.75	884.67	87.28	0.82	4.61	924.09	90.02	0.83	4.56	688.45
	BERT-Attack	79.82	0.87	2.43	**378.79**	93.53	0.88	2.31	**387.95**	92.43	0.88	2.37	**362.54**
	TPGD	**99.88**	**0.91**	**1.01**	531.31	**100**	**0.91**	**-1.35**	787.35	**100**	**0.91**	**1.10**	1020.88
MNLI	Textfooler	72.40	0.83	5.94	780.80	77.30	0.87	5.83	640.21	82.50	0.87	5.68	828.15
	BERT-Attack	87.70	0.87	5.12	484.27	91.30	0.89	5.04	604.22	89.60	0.89	4.92	583.09
	TPGD	**95.65**	**0.91**	**-0.86**	**449.80**	**94.80**	**0.90**	**-0.91**	**331.47**	**96.84**	**0.92**	**-0.93**	**394.49**
AG's News	Textfooler	51.40	0.81	10.86	825.51	98.20	0.95	11.25	372.90	73.20	0.91	11.00	515.87
	BERT-Attack	58.20	**0.91**	10.69	**431.47**	99.30	**0.98**	10.83	307.74	91.30	**0.94**	10.83	418.36
	TPGD	**94.47**	0.75	**-0.18**	614.68	**99.30**	0.88	**1.05**	**258.85**	**99.24**	0.88	**-1.04**	**235.80**

Table 2: The results of automatic evaluation metrics on SST-2, MNLI, and AG's News. ΔI and ΔPPL denotes the increase of grammar errors and perplexity.

5.1 Datasets and Victim Models

For sentiment analysis, we choose SST-2 (Socher et al., 2013), a binary sentiment classification benchmark dataset. For natural language inference, we choose the mismatched MNLI (Williams et al., 2018) dataset. For news classification, we choose AG's News (Zhang et al., 2015) multi-classification datasets with four categories: World, Sports, Business, and Science/Technology. We randomly sample 1,000 samples that models can classify correctly from the test set and perform adversarial attacks on those samples.

For each dataset, we evaluate TPGD by attacking BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2020) with a local fine-tuned BERT model to generate potential adversarial samples. Details of datasets and the original accuracy of victim models are listed in Table 1.

5.2 Baselines

We adopt strong baseline attack models for comparison: **(1) Textfooler** (Jin et al., 2020) is a score-based method that replaces words with their synonyms and enforces a cosine similarity constraint; **(2) BERT-Attack** (Li et al., 2020b) is also a score-based method which employs MLM to find substitute words. We implement these methods with OpenAttack (Zeng et al., 2021).

5.3 Evaluation Metrics

We evaluate our method considering the attack success rate and adversarial samples quality. (1) Attack Success Rate (**ASR**) is the proportion of adversarial samples that successfully mislead victim models' predictions. (2) Quality of adversarial samples is evaluated by two automatic metrics and human evaluation, including their semantic consistency, grammaticality, and fluency. Specifically, we use Universal Sentence Encoder (Cer et al., 2018) to compute the semantic similarity between the original text and the corresponding adversarial sample, Language-Tool[1] to calculate the increase of grammar errors, and GPT-2 (Radford et al., 2019) to compute the perplexity of adversarial samples as a measure of fluency. We also conduct human evaluation to measure the validity and quality of adversarial samples.

5.4 Experimental Results

The results of automatic evaluation metrics are listed in Table 2.

Attack Performance We observe that our method consistently outperforms two strong score-based attack methods considering the attack success rate. We attribute the success of our attack method to the general framework that can easily adapt gradient-based method to textual adversarial attacks and make the search of adversarial samples more effective.

Adversarial Sample Quality Our method can produce adversarial samples with the least number of grammar errors and a low perplexity overall, indicating the high quality of our adversarial samples with better coherence of grammaticality and fluency. Besides, we also observe that our method can produce adversarial samples similar to original samples considering the USE metric. This proves

[1] https://github.com/jxmorris12/language_tool_python

that our method can launch successful adversarial attacks with fewer perturbations.

5.5 Human Evaluations

To further study the quality and validity of adversarial samples, we randomly selected 100 original SST-2 sentences and 100 adversarial samples from Bert-Attack and TPGD respectively. Following Li et al. (2020b), we shuffle the 300 samples and ask 3 independent human judges to evaluate the quality (300 samples per person). For semantic consistency evaluation, we ask humans to predict the labels of mixed texts. For grammar and fluency, human judges score from 1 to 5 on the above examples. All annotators have no knowledge about the source of text, and all their evaluation results are averaged (shown in Table 3).

Semantic consistency. Since human judges have a high accuracy on original text, the prediction results on texts can be regarded as the ground truth semantics. Given that an adversarial example holds the same label with original one, it means ground truth indication agrees with claimed one if humans' prediction consists with the label of adversary, which stands for a good semantic preservation. Therefore, the accuracy can be a criterion for semantic consistency between original sentences and adversarial ones. From the results, human judges achieve an accuracy of 0.68 on TPGD, indicating only 32% of the time our algorithm generates sentences that do actually change the ground truth label. As BERT-Attack scores only 0.48, this result verifies that the adversarial samples crafted by TPGD have a better semantic consistency.

Grammar and Fluency. Both BERT-Attack and TPGD suffer a decline in grammatical correctness and fluency of adversarial text, but TPGD still performs slightly better than BERT-Attack.

Source	Accuracy	Grammar & Fluency
Original	0.92	4.63
BERT-Attack	0.48	3.41
TPGD	0.68	3.52

Table 3: Human Evaluation on SST-2 with RobERTa in terms of prediction accuracy, grammar correctness and fluency.

6 Conclusion and Future Work

In this paper, we propose a general framework to adapt gradient-based adversarial attack methods investigated in CV to NLP. In our framework, the problem of searching textual adversarial samples is transformed from the discrete text space to the embedding layer, where continuous gradient-based perturbations can be directly added to. The perturbations will be amplified in the forward propagation process. Then a MLM head is employed to decode the final perturbed latent representations. With its extensive linguistic knowledge, the coherence and grammaticality of the adversary samples can be guaranteed. We instantiate our framework with TPGD, including the iterative perturbation process and the reconstruction process, to perform decision-based black-box attack. We conduct exhaustive experiments to evaluate our framework and TPGD algorithm. Experimental results show the superiority of our method, especially in terms of attack success rate and adversarial samples quality.

In the future, we will adapt other gradient-based methods in CV with our framework and explore to improve models' robustness through adversarial training.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61602197.

Ethical Consideration

In this section, we discuss the potential broader impact and ethical considerations of our paper.

Intended use. In this paper, we design a general framework to adapt existing gradient-based methods in CV to NLP, and further propose a decision-based textual attack method with impressive performance. Our motivations are twofold. First, we attempt to introduce adversarial attack methods of CV to NLP, since image attack methods have been well-explored and proved to be effective, therefore helping this two fields better share research resources hence accelerate the research process on both sides. Second, we hope to find insights about interpretability and robustness of current black-box DNNs from our study.

Potential risk. There is a possibility that our attack methods may be used maliciously to launch adversarial attacks against off-the-shelf commercial systems. However, studies on adversarial attacks are still necessary since it is important for

the research community to understand these powerful attack models before defending against these attacks.

Energy saving. We list the settings of hyperparameters of our method in Appendix, to prevent people from conducting unnecessary tuning and help researchers to quickly reproduce our results. We will also release the checkpoints including all victim models to avoid repeated energy costs.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, Mahdieh Soleymani Baghshah, and Pascal Frossard. 2019. Universal adversarial attacks on text classifiers. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7345–7349.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Yangyi Chen, Jin Su, and Wei Wei. 2021. Multi-granularity textual adversarial attack with behavior cloning. *arXiv preprint arXiv:2109.04367*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.

Cheng Chi, Shifeng Zhang, Junliang Xing, Zhen Lei, Stan Z. Li, and Xudong Zou. 2019. Selective refinement network for high performance face detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8231–8238.

Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. 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)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Christiane Fellbaum. 2010. *WordNet*, pages 231–243. Springer Netherlands, Dordrecht.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment.

Wei-Tsung Kao, Tsung-Han Wu, Po-Han Chi, Chun-Cheng Hsieh, and Hung-Yi Lee. 2021. Bert's output layer recognizes all hidden layers? some intriguing phenomena and a simple way to boost bert.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. *Proceedings 2019 Network and Distributed System Security Symposium*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020a. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. Bert-attack: Adversarial attack against bert using bert.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards deep learning models resistant to adversarial attacks.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021. Generating natural language attacks in a hard label black box setting.

Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016a. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 372–387.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016b. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 372–387.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016c. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021. Mind the style of text! adversarial and backdoor attacks based on text style transfer. *arXiv preprint arXiv:2110.07139*.

Alec Radford, Jeffrey Wu, and Rewon Child. 2019. Rewon child, david luan, dario amodei, and ilya sutskever. 2019. *Language models are unsupervised multitask learners. OpenAI Blog*, 1(8):9.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Andrew Ross and Finale Doshi-Velez. 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. 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)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Jincheng Xu and Qingfeng Du. 2020. Texttricker: Loss-based and gradient-based adversarial attacks on text classification models. *Engineering Applications of Artificial Intelligence*, 92:103641.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020.

Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online. Association for Computational Linguistics.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. OpenAttack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 363–371, Online. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.