Towards Semantics- and Domain-Aware Adversarial Attacks

Jianping Zhang[1] , **Yung-Chieh Huang**[2] , **Weibin Wu**[3] * and **Michael R. Lyu**[1]

[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2]Department of Computer Science, University of Illinois Urbana-Champaign
[3]School of Software Engineering, Sun Yat-sen University

{jpzhang, lyu}@cse.cuhk.edu.hk, ych10@illinois.edu, wuwb36@mail.sysu.edu.cn

Abstract

Language models are known to be vulnerable to textual adversarial attacks, which add human-imperceptible perturbations to the input to mislead DNNs. It is thus imperative to devise effective attack algorithms to identify the deficiencies of DNNs before real-world deployment. However, existing word-level attacks have two major deficiencies: (1) They may change the semantics of the original sentence. (2) The generated adversarial sample can appear unnatural to humans due to the introduction of out-of-domain substitute words. In this paper, to address such drawbacks, we propose a semantics- and domain-aware word-level attack method. Specifically, we greedily replace the important words in a sentence with the ones suggested by a language model. The language model is trained to be semantics- and domain-aware via contrastive learning and in-domain pre-training. Furthermore, to balance the quality of adversarial examples and the attack success rate, we propose an iterative updating framework to optimize the contrastive learning loss and the in-domain pre-training loss in circular order. Notably, compared with state-of-the-art benchmarks, our strategy can achieve over 3% improvement in attack success rates and 9.8% improvement in the quality of adversarial examples.

1 Introduction

Deep neural networks (DNN) have been deployed in many real-world applications, such as machine translation [Stahlberg, 2020; Wang *et al.*, 2022] and sentiment analysis [Birjali *et al.*, 2021]. However, recent research shows that DNNs are vulnerable to adversarial attacks [Zhang *et al.*, 2022; Zhang *et al.*, 2023a; Zhang *et al.*, 2023b; Szegedy *et al.*, 2013; Jin *et al.*, 2020; Liu *et al.*, 2022a; Liu *et al.*, 2022b], which add human-imperceptible perturbations to the input to mislead DNNs. It is thus imperative to devise effective attack algorithms to identify the deficiencies of DNNs before deployment, which serves as the first step to improve their robustness [Wang *et al.*, 2023]. However, due to the discrete and non-differentiated nature of the text space, it is challenging to craft textual adversarial examples [Morris *et al.*, 2020].

There are mainly four kinds of textual adversarial attacks based on the granularity of modification: character-level, word-level, sentence-level, and multi-level [Huq *et al.*, 2020; Qiu *et al.*, 2022]. Character-level attacks generally insert, delete, flip, replace, or swap individual characters in the text. Word-level attacks generally add new words, remove words, or change words in the sentences. Sentence-level attacks usually insert new sentences or paraphrase the original sentences. Multi-level attacks combine character-level, word-level, and sentence-level attacks together to craft adversarial examples. Compared with sentence-level attacks, character-level and word-level attacks have higher attack success rates [Zeng *et al.*, 2021]. Besides, word-level attacks are more stealthy than character-level attacks, which often introduce typos [Ebrahimi *et al.*, 2018]. Therefore, in this work, we focus on word-level textual attacks.

As demonstrated in Table 1, existing word-level adversarial attacks suffer from two primary pitfalls [Jin *et al.*, 2020; Ren *et al.*, 2019; Li *et al.*, 2020]: (1) They often fail to preserve the semantic meanings of the original sentences. For example, BERT-Attack replaces "top" with "bottom", which changes the semantics of the original sentence. (2) They can introduce substitute words that are out of the domain of the original sentences, resulting in unnatural sentences. For example, BAE replaces "performances" with "finances", which transforms the domain of the original sentence from movie reviews to finance to some extent. However, the transformation is inconsistent with the overall context of the original sentence. Therefore, the quality of adversarial examples crafted by state-of-the-art approaches is unsatisfactory. Although multiple rules of regularization have been proposed to improve the quality of adversarial examples, like utilizing a grammar checker to reduce grammatical errors and constraining the substitute words to be synonyms of the target words [Ren *et al.*, 2019], such strategies still cannot address the above two deficiencies at the same time.

In this paper, we aim to solve the aforementioned issues from a data-driven point of view. To this end, we propose a language model-based word-level attack method. Specifically, we first identify words that are crucial for the pre-

*Corresponding author.

Method	Text	Issue
Original	... rivals the top japanese animations of recent vintage .	Inconsistent
BERT-Attack	... rivals the bottom japanese animations of recent vintage .	Semantics
Ours	... rivals the high japanese animations of recent vintage .	
Original	the performances are immaculate , with roussillon providing comic relief .	Out-of-domain
BAE	the finances are immaculate , with roussillon providing comic relief .	Replacement
Ours	the script are immaculate , with roussillon providing comic relief .	

Table 1: Qualitative comparison of generated adversarial examples. Only words in blue are perturbed. Overlong text is cut to fit in the table.

dictions of DNNs. Then instead of searching from a large vocabulary, we train a language model to produce candidate substitute words. The language model is trained to be semantics- and domain-aware via contrastive learning and in-domain pre-training. As a result, the incurred replacement is semantics-preserving and relevant to the original domain of the sentences. To generate both high-quality and effective adversarial samples, we further develop an iterative updating framework to balance the quality of adversarial examples and the attack success rates, which iteratively optimizes the contrastive learning loss and the in-domain pre-training loss during the training of our language model. We finally greedily replace the important words in a sentence with the ones suggested by the trained language model to craft adversarial sentences.

The contributions of this paper are three-fold:

- We propose a data-driven method based on contrastive learning and in-domain pre-training to solve the changed semantics and the out-of-domain replacement problems of existing word-level adversarial attacks.
- To generate both high-quality and effective adversarial samples, we balance the quality and attack success rates of adversarial samples via developing an iterative updating framework to properly combine contrastive learning and in-domain pre-training.
- Comprehensive experiments confirm the superiority of our approach over state-of-the-art baselines in both the attack success rates and the quality of generated adversarial samples.

2 Related Work

Current textual adversarial attacks generally have four categories based on the granularity of the incurred perturbations: character-level, word-level, sentence-level, and multi-level.

Character-level adversarial attacks perturb characters regardless of the spellings and grammatical rules [Ebrahimi *et al.*, 2018]. The modifications generally include inserting, deleting, flipping, replacing, or swapping individual characters in the text. However, such modifications can produce unnatural adversarial samples, which are easily noticeable and can hardly bypass a grammar checker.

Sentence-level adversarial attacks usually insert new irrelevant sentences [Jia and Liang, 2017] or paraphrase the original sentences [Iyyer *et al.*, 2018]. Although the adversarial examples are natural to humans, the attack success rates are lower than character-level and word-level attacks [Zeng *et al.*, 2021].

Word-level adversarial attacks generally add new words, remove words, or change words in the sentences. They are not only more effective than sentence-level adversarial attacks but also more stealthy than character-level adversarial attacks. Therefore, in this work, we focus on word-level adversarial attacks.

Multi-level attacks combine character-level, word-level, and sentence-level attacks together to craft adversarial examples [Chen *et al.*, 2021], which still suffer from the deficiencies of these single-level attacks and can incur larger perturbations than these single-level attacks.

Word-level adversarial attacks usually involve two steps: finding the candidate substitute words and searching for adversarial examples. Candidate substitute words can be the ones with similar word embeddings to the target words, synonyms of the target words [Miller, 1998], or the words suggested by a language model [Kenton and Toutanova, 2019]. The method of searching for adversarial examples includes greedy search algorithms [Ren *et al.*, 2019], genetic algorithms [Alzantot *et al.*, 2018], and particle swarm optimization [Zang *et al.*, 2020].

In this paper, we focus on language model-based word-level attacks. Compared with other strategies to find the candidate substitute words, a language model-based word-level attack takes the context into consideration, which is conducive to generating natural adversarial samples. However, existing language model-based word-level attacks suffer from the problem of semantic changes and out-of-domain replacement [Jin *et al.*, 2020; Ren *et al.*, 2019; Li *et al.*, 2020]. To overcome such drawbacks, we first propose to train a semantics- and domain-aware language model via contrastive learning and in-domain pre-training, which can produce semantics-preserving and in-domain replacement words. Besides, to balance the quality of adversarial examples and the attack success rates, we then propose an iterative updating framework to properly combine contrastive learning and in-domain pre-training. We finally search for adversarial samples via a greedy search algorithm. Consequently, our method can generate more high-quality and effective adversarial sentences than prior efforts.

3 Methodology

In this work, we propose a language model-based word-level attack method. We attempt to resolve the issues of changed semantics and out-of-domain replacement from a data-driven point of view. In short, we utilize contrastive learning to train a semantics-aware language model to produce semantics-preserving replacements (Section 3.1). We also employ in-

domain pre-training to train a domain-aware language model to output in-domain replacement (Section 3.1). To generate both high-quality and effective adversarial samples, we then propose an iterative updating framework to properly combine contrastive learning and in-domain pre-training (Section 3.2). We finally employ the trained semantics- and domain-aware language model to craft adversarial samples via a greedy search algorithm (Section 3.3).

3.1 Semantics- and Domain-Aware Language Model

Semantics-Aware Language Model

A language model can predict candidate words to fill in a masked sentence. However, given a target sentence, the distance between a similar meaning sentence and an opposite meaning sentence can be small in the embedding space of a language model, making it hard to separate them apart. Therefore, the language model may produce candidate words that cannot preserve the original meaning of a target sentence. In order to generate semantics-aware replacement, the language model should be sensitive to semantic changes.

We get inspiration from the contrastive learning methodology [Gao *et al.*, 2021], which pushes away the sentences with opposite meanings and pulls close the sentences with similar meanings in the representation space to learn a better sentence embedding. In short, we design a contrastive learning algorithm for generating semantics-aware replacement under the adversarial attack scenario. The key is to train a language model to distinguish synonyms and antonyms in the embedding space. We present how to construct contrastive examples under the adversarial attack scenario and the training objective of the semantics-aware language model as follows.

Contrastive Examples. The contrastive examples should help to train the language model to be sensitive to semantic changes. Therefore, the contrastive examples should consist of sentences with both similar and opposite semantics. Furthermore, since attackers can only perturb a few words under the setting of adversarial attacks, the difference between a pair of contrastive examples should be small.

Therefore, given an original sentence, we generate a pair of contrastive examples, where one of them is semantically similar to the original sentence, while the other is semantically different from the original sentence. The semantically similar example is generated by replacing the representative words in the original sentence with their synonyms, hypernyms, and morphological neighbors. In contrast, the semantically different example is constructed by replacing the same words in the original sentence with their antonyms. The replacement words are from WordNet [Miller, 1998]. The representative words should contain semantic information, which includes verbs, nouns, adjectives, and adverbs. To make the difference between the contrastive pair and the original sentence small, the number of replaced words is small, which is about 20% of the representative words in the original sentences.

Training Objective. We define the combination of a contrastive pair and the original sentence as (x_i^+, x_i^-, x_i), where x_i^+, x_i^-, and x_i are the semantically similar example, the semantically different example, and the original example, respectively. The goal of contrastive learning is to pull close the semantically similar examples while pushing away semantically different examples in the feature space. Therefore, we define the loss function of contrastive learning based on the supervised version of Simcse [Gao *et al.*, 2021]. The training objective is shown below:

$$L_{sem} = -log \frac{e^{f(h_i, h_i^+)/\tau}}{\sum_{j=1}^{N} (e^{f(h_i, h_j^+)/\tau} + e^{f(h_i, h_j^-)/\tau})}. \quad (1)$$

N is the batch size. h_i^+, h_i^-, and h_i are the embeddings of the semantically similar example, the semantically different example, and the original example generated by the language model, respectively. The function $f(a, b) = \frac{a^T b}{\|a\|_2 \|b\|_2}$ computes the cosine similarity of two feature vectors. τ is a hyper-parameter that we set to be 0.05.

The differences between our method with other contrastive learning methods are two-fold. First, we employ contrastive learning to train a language model to generate the semantics-aware replacement. Second, we construct contrastive examples by changing a small ratio of representative words to their synonyms or antonyms, which fits the scenario of adversarial attacks.

Domain-Aware Language Model

There can be a gap between the original pre-training domain of the language model and the target domain of the victim model. For example, the language model is pre-trained on a corpus of Wikipedia, while the victim model is designed for analyzing movie reviews. Therefore, if we directly employ an off-the-shelf pre-trained language model like previous adversarial attack methods, the generated candidate words can be out-of-domain. For example, as shown in Table 1, replacing "performances" with "finances" satisfies grammatical rules. However, the replacement word comes from the domain of finance, while the original sentence is in the domain of movie reviews. Therefore, the replacement is not consistent with the overall context of the original sentence, which results in unnatural sentences. To solve this issue, we propose to train a domain-aware language model, which can produce in-domain replacement words. In short, we restart pre-training [Gururangan *et al.*, 2020] under the adversarial attack scenario to reduce the domain gap between the language model and the victim model. We describe how to construct the pre-training examples and the training objective of the domain-aware language model as follows.

Pre-training Examples. We utilize the datasets whose domain is similar to that of the victim model's training data to pre-train the language model. For example, if we are going to attack a sentiment analysis model trained on the movie review dataset MR, we can pre-train the domain-aware language model on datasets of similar domains, like IMDB or SST-2. After pre-training on such related datasets, the language model can generate in-domain replacement words for adversarial attacks. Notably, if we adopt the same training set of the victim model to pre-train the language model, the generated adversarial sentence will be close to the original training data of the victim model, hurting the attack success rates. Therefore, we use related datasets for pre-training instead of the original training set of the victim model.

Datasets	MR	IMDB	SST-2	MNLI	SNLI
BiLSTM	78.1	74.0	84.5	63.8	69.2
BERT	83.8	94.3	92.4	84.0	89.7
DistilBERT	83.4	93.7	90.0	81.7	87.0

Table 2: The accuracy (%) of the pre-trained victim models.

Training Objective. Following BERT [Kenton and Toutanova, 2019], we adopt the training objective L_{mlm} of a masked language model (MLM) to train the domain-aware language model. Specifically, given a training sentence, we train the domain-aware language model to predict a random sample of input tokens in the training sentence that have been replaced by a [MASK] placeholder in a multi-class setting. By doing so, the domain gap between the trained language model and the victim model can be reduced. Therefore, the resultant language model can generate in-domain replacement word that is consistent with the overall context of the target sentence.

3.2 Iterative Updating Framework

In the previous sections, we utilize the techniques of contrastive learning and in-domain pre-training to make the trained language model semantics-aware and domain-aware, respectively. In this section, we present how to combine these two techniques properly to generate both high-quality (i.e., semantics-aware and domain-aware) and effective (i.e., error-inducing) adversarial samples.

A straightforward solution is to optimize the two losses (L_{sem} and L_{mlm}) at the same time when training a language model. However, in this way, the trained language model can easily overfit and become extremely sensitive to changed semantics and out-of-domain replacement. As a result, the diversity of the candidate replacement words predicted by the language model will drop dramatically, which results in only paraphrasing the original sentences, instead of generating error-inducing adversarial samples.

Therefore, to generate both high-quality and effective adversarial samples, we propose an iterative updating framework to train the language model, which can combine the semantic loss L_{sem} and the masked language model loss L_{mlm} properly. The key is to balance the quality and diversity of the candidate replacement words predicted by the trained language model. To this end, we propose to introduce some variances into the training of the language model, which is achieved by optimizing the two losses in circular order during model training. As a result, it can avoid overfitting to the two objectives and improve the diversity of the candidate replacement words predicted by the trained language model.

Specifically, our iterative updating framework proceeds as follows: (1) We randomly split the whole contrastive examples C and pre-training examples P into T parts. We denote the i-th split of the contrastive examples and pre-training examples as C_i and P_i, respectively. (2) In the i-th iteration, we first update the language model with the semantic loss L_{sem} on the contrastive example split C_i. Then we update the language model with the masked language model loss L_{mlm} on the pre-training example split C_i. (3) We iteratively train the language model for T cycles by repeating Step 2.

3.3 Search Algorithm

In this section, we present the search algorithm to generate adversarial samples based on our semantics- and domain-aware language model. There are two steps in our search algorithm: finding the important words and greedily replacing the important words until a successful attack or achieving the query budget.

Important Words. Following the previous importance-based adversarial attacks [Li *et al.*, 2020], we define the importance of a word by the confidence score reduction of the victim model when we mask out the word. We denote the input sequence as $S = [w_0, \cdots, w_i, \cdots]$, and the input sequence with the word w_i masked out as $S_{\setminus w_i} = [w_0, \cdots, w_{i-1}, [MASK], w_{i+1} \cdots]$. For simplicity, we just delete the word from the sentence without a mask token. Therefore, the importance of the word w_i is:

$$I_{w_i} = O_y(S) - O_y(S_{\setminus w_i}), \tag{2}$$

where $O_y(S)$ is the model's confidence score of classifying the input sequence S into the true label y.

Word Replacement. Based on the importance of words in a target sentence, we replace them one by one in descending order to generate an adversarial example S_{adv}. Specifically, in each iteration, we mask out the current important token and utilize the trained semantics- and domain-aware language model to predict the masked token. We then replace the current important token with the prediction of the language model. Besides, following [Jin *et al.*, 2020], we utilize Universal Sentence Encoder (USE) to regularize sentence similarity and Part of Speech (POS) to correct the grammar. We replace the important tokens in the target sentence one by one until the perturbed sentence causes misclassification or the maximum query budget is achieved. If no replacement causes misclassification, we choose the replacement causing the most reduction in the model's confidence score of classifying the perturbed sentence into the ground-truth label.

Comparison with BERT-Based Attacks. The most important difference is that we train a semantics- and domain-aware language model to generate candidate substitute words, which solves the issues of changed semantics and out-of-domain replacement in a data-driven way. Furthermore, our iterative updating framework balances the quality and diversity of generated adversarial samples. As a result, we can produce both high-quality and effective adversarial sentences.

4 Experiment

4.1 Experimental Setup

Datasets. We compare our approach with state-of-the-art baselines on different text classification tasks: sentiment analysis and natural language inference. For the sentiment analysis task, we choose MR [Pang and Lee, 2005], IMDB [Maas *et al.*, 2011], and SST-2 [Socher *et al.*, 2013], which are widely used datasets tailored for binary sentiment classification. For the natural language inference task, we select

Dataset	Attack	BiLSTM			BERT			DistilBERT		
		ASR	Query	PP	ASR	Query	PP	ASR	Query	PP
MR	PWWS	61.5	69.7	20.0	51.8	62.4	16.0	57.3	70.2	19.2
	TextFooler	81.7	58.6	11.5	63.4	58.4	20.8	63.3	62.4	13.0
	BAE	68.0	57.4	11.8	56.4	63.9	13.5	61.5	59.7	12.2
	BERT-Attack	70.3	67.6	10.7	55.4	58.6	12.2	64.0	65.1	10.3
	Ours	**82.3**	**53.5**	**10.3**	**65.4**	**57.6**	**11.8**	**69.4**	**54.3**	**10.1**
IMDB	PWWS	54.8	397	4.5	39.3	355	5.5	54.4	397	4.5
	TextFooler	73.4	289	2.5	49.2	331	3.6	68.3	296	2.8
	BAE	58.9	288	3.0	43.3	325	4.1	67.8	376	3.4
	BERT-Attack	66.0	298	**2.2**	42.4	**315**	**2.4**	65.1	293	2.0
	Ours	**75.7**	**265**	**2.2**	**59.5**	331	**2.4**	**73.6**	**270**	**1.6**
SST-2	PWWS	64.0	68.7	19.0	54.0	59.7	16.4	59.1	69.6	21.4
	TextFooler	**76.0**	59.8	13.0	68.0	52.4	21.0	67.1	61.9	13.3
	BAE	68.0	56.5	12.4	58.6	60.8	12.9	60.8	59.2	12.8
	BERT-Attack	68.3	65.8	11.4	64.8	68.8	11.1	67.0	65.8	10.8
	Ours	75.4	**51.7**	**10.7**	**68.7**	**50.4**	**10.9**	**70.1**	**51.8**	**10.6**
MNLI	PWWS	57.2	73.3	9.5	69.6	66.0	9.0	64.1	76.7	11.6
	TextFooler	75.3	63.5	5.5	83.6	**58.3**	8.3	76.6	62.8	7.0
	BAE	73.9	63.9	5.7	80.2	61.8	6.9	77.9	**62.1**	6.9
	BERT-Attack	80.2	70.0	**5.0**	86.3	67.9	5.5	84.8	70.3	**5.4**
	Ours	**81.7**	**63.0**	5.2	**87.3**	61.3	**5.4**	**86.4**	62.8	**5.4**
SNLI	PWWS	77.8	74.0	9.7	77.6	57.0	11.4	76.8	74.2	10.9
	TextFooler	93.3	56.8	6.6	**92.9**	54.3	7.1	90.9	53.4	7.0
	BAE	83.6	**56.2**	6.5	76.9	52.9	7.4	76.1	**52.3**	7.6
	BERT-Attack	93.7	66.3	**5.9**	91.0	64.7	6.1	91.2	66.3	5.9
	Ours	**94.3**	58.3	**5.9**	**92.9**	**52.1**	**5.9**	**91.8**	**52.3**	**5.8**

Table 3: The performance of different attacks against three victim models (BiLSTM, BERT, and DistilBERT) trained on different datasets. The best result is in bold.

MNLI [Williams *et al.*, 2018] and SNLI [Bowman *et al.*, 2015] datasets.

Victim Models. We evaluate attacking methods by attacking three victim models, including BiLSTM [Schuster and Paliwal, 1997], BERT [Kenton and Toutanova, 2019], and DistilBERT [Sanh *et al.*, 2019]. The weights of the victim models are publicly available at the TextAttack [Morris *et al.*, 2020] package. Since the weights of BiLSTM on MNLI and SNLI are not available, we fine-tune the model on the training sets of MNLI and SNLI, respectively. The accuracy of all the victim models on the clean test set is shown in Table 2.

Baselines. We implement the baselines with the open-source NLP adversarial attack packages TextAttack [Morris *et al.*, 2020] and OpenAttack [Zeng *et al.*, 2021]. We compare our approach with four representative importance-based methods, which perturb the original sentence based on the word importance: (1) TextFooler [Jin *et al.*, 2020] ranks words in the original sentence by saliency and chooses substitute words based on the word embedding to construct an adversarial example. (2) PWWS [Ren *et al.*, 2019] utilizes augmented word saliency to replace words with synonyms from WordNet [Miller, 1998] iteratively in a greedy manner. (3) BAE replaces the important words in the original sentence with the ones predicted from MLM until the resultant adversarial example misleads the victim model. (4) BERT-Attack [Li *et al.*, 2020] generates replacement words from MLM by masking the important words in the original sentence. It then greedily selects the replacement words that can cause the largest drop in the victim model's confidence score for the ground-truth label.

Metrics. We evaluate the performance of attacking methods by three metrics: (1) ASR (Attack Success Rate) is the ratio of the adversarial examples that successfully mislead the victim model among all the generated adversarial examples. (2) Query is the average query number of an attacking approach to successfully craft the adversarial example, which reflects the efficiency of the attacking method. (3) PP (Perturbed Percentage) is the percentage of the perturbed words in the target sentence, which shows the semantic consistency in general, since fewer perturbations usually imply better semantic consistency.

In order to evaluate the quality of the generated adversarial sample, we first utilize automatic measures to assess the semantic consistency and naturalness of a sentence. Specifically, we follow [Huang *et al.*, 2022] to compute the Sem score and the Syn score to measure the semantic consistency and naturalness of a sentence, respectively. The Sem score combines the Levenshtein distance [Levenshtein and others, 1966] and the sentence embedding model to evaluate the semantic similarity. The Syn score computes the reciprocal of the perplexity of PLM, which reflects the naturalness of a sentence. For the PLM, we choose RoBERTa [Liu *et al.*, 2019]. We also ask human annotators to score the quality of adversarial examples from the two aspects.

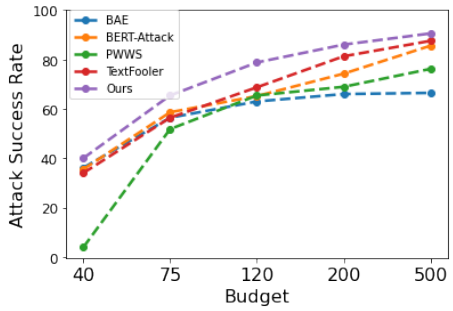

Figure 1: The ASR results under different query budgets.

Datasets	Attack	BiLSTM	DistilBERT
		ASR	ASR
MR	PWWS	9.9	15.8
	TextFooler	12.3	15.6
	BAE	13.9	24.5
	BERT-Attack	8.1	19.9
	Ours	**18.6**	**31.0**
SST-2	PWWS	11.8	5.7
	TextFooler	17.5	4.3
	BAE	21.5	12.6
	BERT-Attack	16.6	8.1
	Ours	**26.6**	**19.8**
MNLI	PWWS	20.5	24.3
	TextFooler	34.3	27.9
	BAE	35.5	35.8
	BERT-Attack	36.8	39.4
	Ours	**41.9**	**39.9**

Table 4: The performance of different attacks in the transfer attack setting by attacking a BERT model. The best result is in bold.

Parameter Settings. We utilize the default hyperparameters of baselines to test their performance under a fixed query budget. In order to show the efficiency and effectiveness of different attacking methods, we set the maximum query number to be 300 for the IMDB dataset and 75 for the other datasets due to the difficulty of the IMDB dataset.

For our semantics- and domain-aware language model, we choose pre-trained BERT as the architecture. Besides, we consider the top 20 synonyms from its predictions as the candidate substitute words. We set a threshold of 0.8 for the cosine similarity between USE-based embeddings of the adversarial example and the original input sentence. We make use of task-related training datasets to construct contrastive examples and pre-training examples to train our language model. For example, for the sentiment analysis task, we use the IMDB dataset to train our language model for attacking models fine-tuned on the MR dataset. For the natural language inference task, we use the SNLI dataset to train our language model for attacking models fine-tuned on the MNLI.

4.2 Attacking Performance

Table 3 shows the performance of different attacking methods. Our approach consistently outperforms the state-of-the-art baselines in all the settings. Our approach improves the attack success rate by an average margin of 3% with 9% fewer queries on average. We further compare the performance of different attacking methods on the MR dataset under different query budgets in Figure 1. Our approach still consistently outperforms the baselines under all budget settings.

In addition, compared with the state-of-the-art baselines, our approach can generate more semantically consistent and in-domain adversarial sentences with respect to the original sentences. Table 1 shows some representative examples: (1) Our method replaces "top" with "high" in the first sentence, which preserves the semantic meaning of the original sentence. (2) Our method replaces "performances" with "script" in the second sentence, which falls into the domain of the original sentence (i.e., movie reviews).

4.3 Transferability

We further conduct experiments to study the transferability of different attacking methods. We set BERT as the white-box model. Table 4 shows the results. Our attacking method achieves the highest attack success rates in all settings, compared to the state-of-the-art baselines. Therefore, it confirms that the adversarial sentences generated by our attacking method have good transferability.

4.4 Quality of Adversarial Examples

Table 5 compares the quality of adversarial examples generated by different attacking methods. We can see that our approach consistently outperforms the state-of-the-art baselines, in terms of both semantic consistency (the Sem score) and naturalness (the Syn score). Notably, our approach improves the Sem score with a large margin of 9.8 %.

4.5 Candidate Substitute Words

To examine why our attacking approach can generate both high-quality and effective adversarial sentences, we show the candidate substitute words output by different attacking schemes in Table 6. We can see that the top-10 candidate substitute words of our approach are more in-domain with the original sentence. For example, the candidate substitute words generated by BAE include "billing", which is out of the domain of the original sentence. Furthermore, the candidate substitute words generated by our approach without the iterative updating framework are not diverse and are constrained to the synonym of "performance" or "award". In contrast, the candidate substitute words of our approach cover "actor", "effect", "performance", and "director", which are more diverse. We expect that high diversity of generated candidate substitute words can contribute to high attack success rates, which is confirmed in our ablation study.

4.6 Ablation Study

Different Components. We examine the effectiveness of different components in our attacking method by attacking the BERT model trained on the MR dataset. Table 7 shows the results. "Init" means generating the candidate substitute words with the vallina BERT. "L_{mlm} (MNLI)" means training BERT with only the L_{mlm} loss on the task-unrelated dataset (MNLI), while "L_{mlm} " means training BERT with only the L_{mlm} loss on the task-related dataset (MR). "$L_{sem}+$

Dataset	Victim Model	BiLSTM		BERT		DistilBERT	
	Attacks	Syn	Sem	Syn	Sem	Syn	Sem
MR	PWWS	0.123	0.567	0.171	0.738	0.119	0.543
	TextFooler	0.144	0.633	0.169	0.737	0.140	0.598
	BAE	0.146	0.600	0.163	0.654	0.143	0.583
	BERT-Attack	0.144	0.624	0.165	0.686	0.142	0.599
	Ours	**0.151**	**0.647**	**0.176**	**0.792**	**0.147**	**0.742**
SST-2	PWWS	0.118	0.577	0.134	0.504	0.122	0.547
	TextFooler	0.150	0.660	0.139	0.635	0.144	0.589
	BAE	0.147	0.564	0.140	0.527	0.141	0.529
	BERT-Attack	0.148	0.623	0.139	0.581	0.141	0.588
	Ours	**0.153**	**0.670**	**0.144**	**0.707**	**0.145**	**0.727**
MNLI	PWWS	0.161	0.770	0.167	0.741	0.146	0.741
	TextFooler	0.185	0.885	0.178	0.932	0.180	0.835
	BAE	0.185	0.864	**0.180**	0.902	0.182	0.849
	BERT-Attack	0.183	0.937	0.179	0.933	0.181	0.920
	Ours	**0.188**	**0.957**	**0.180**	**0.948**	**0.188**	**0.937**

Table 5: The quality of adversarial sentences generated by different attacking methods. The best result is in bold.

Method	Top-10 Candidates
Original	the performances are immaculate , with roussillon providing comic relief .
BAE	performance, time, credit, work, role, performances, impression, experience, job, billing
Ours w/o Iterative	performance, job, role, award, oscar, career, performances, title, awards, debut
Ours	characters, actors, effects, scenes, films, acting, performances, movies, director, directing

Table 6: Candidate substitute words output by different attacking methods. Only words in blue are perturbed.

Methods	ASR	Syn	Sem
Init	56.4	0.164	0.654
L_{mlm} (MNLI)	57.1	0.141	0.677
L_{mlm}	56.7	**0.181**	0.688
$L_{sem} + L_{mlm}$	60.6	0.179	**0.848**
Ours	**65.4**	0.176	0.792

Table 7: Effectiveness of different components in our attacking method. The best result is in bold.

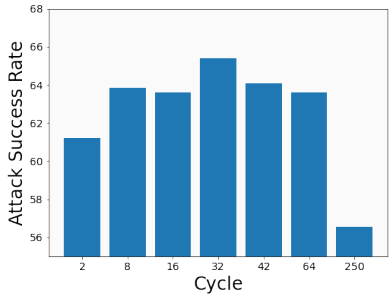

Figure 2: The effect of the number of training cycles.

L_{mlm}" means training the language model by directly optimizing the L_{sem} loss and the L_{mlm} loss together, without the iterative updating.

Training BERT with contrastive learning and in-domain pre-training can improve the attacking performance and the quality of the generated adversarial samples, compared to the "Init" baseline. Besides, using task-related datasets in in-domain pre-training can help to improve the quality of the generated adversarial samples compared to using task-unrelated datasets. Combining contrastive learning and in-domain pre-training with our iterative updating framework achieves the best attacking performance, while the quality of the generated adversarial samples is dropped a little, compared to the "$L_{sem} + L_{mlm}$" baseline. It confirms the effectiveness of our iterative updating framework to balance the attacking performance and the sample quality.

Number of Training Cycles. To examine the effect of the number of training cycles on the trained language model, we attack the BERT model trained on the MR dataset with the language model trained with different numbers of training cycles. As shown in Figure 2, we can see that the ASRs increase as the number of training cycles increases first. However, the ASRs start to decrease after the number of training cycles is larger than 32. It means that the trained language model may underfit with small training cycles while overfit with large training cycles. Therefore, we should choose a moderate number of training cycles to achieve the best ASRs.

5 Conclusion

Word-level attacks suffer from two major pitfalls: inconsistent semantics and out-of-domain replacement. We propose to employ contrastive learning and in-domain pre-training to solve the issues. We further propose an iterative updating framework for generating both high-quality and effective adversarial examples. Experiments corroborate that our method can outperform the SOTA baselines by a considerable margin.

Acknowledgments

The work described in this paper was supported by the National Natural Science Foundation of China (Grant No. 62206318) and the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 14206921 of the General Research Fund).

References

[Alzantot *et al.*, 2018] 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*, pages 2890–2896, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[Birjali *et al.*, 2021] Marouane Birjali, Mohammed Kasri, and Abderrahim Beni-Hssane. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226:107134, 2021.

[Bowman *et al.*, 2015] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[Chen *et al.*, 2021] Yangyi Chen, Jin Su, and Wei Wei. Multi-granularity textual adversarial attack with behavior cloning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4511–4526, 2021.

[Ebrahimi *et al.*, 2018] Javid Ebrahimi, Daniel Lowd, and Dejing Dou. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, 2018.

[Gao *et al.*, 2021] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, 2021.

[Gururangan *et al.*, 2020] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics.

[Huang *et al.*, 2022] Jen-tse Huang, Jianping Zhang, Wenxuan Wang, Pinjia He, Yuxin Su, and Michael R. Lyu. Aeon: A method for automatic evaluation of nlp test cases. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2022, page 202–214, New York, NY, USA, 2022. Association for Computing Machinery.

[Huq *et al.*, 2020] Aminul Huq, Mst Pervin, et al. Adversarial attacks and defense on texts: A survey. *arXiv preprint arXiv:2005.14108*, 2020.

[Iyyer *et al.*, 2018] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 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, June 2018. Association for Computational Linguistics.

[Jia and Liang, 2017] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[Jin *et al.*, 2020] 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*, volume 34, pages 8018–8025, 2020.

[Kenton and Toutanova, 2019] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[Levenshtein and others, 1966] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[Li *et al.*, 2020] 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)*, pages 6193–6202, 2020.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[Liu *et al.*, 2022a] Zihan Liu, Yun Luo, Lirong Wu, Zicheng Liu, and Stan Z Li. Towards reasonable budget allocation in untargeted graph structure attacks via gradient debias. In *Advances in Neural Information Processing Systems*, 2022.

[Liu *et al.*, 2022b] Zihan Liu, Yun Luo, Zelin Zang, and Stan Z Li. Surrogate representation learning with isometric mapping for gray-box graph adversarial attacks. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 591–598, 2022.

[Maas *et al.*, 2011] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for*

computational linguistics: Human language technologies, pages 142–150, 2011.

[Miller, 1998] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[Morris *et al.*, 2020] John Morris, Jin Yong Yoo, and Yanjun Qi. Textattack: Lessons learned in designing python frameworks for nlp. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 126–131, 2020.

[Pang and Lee, 2005] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[Qiu *et al.*, 2022] Shilin Qiu, Qihe Liu, Shijie Zhou, and Wen Huang. Adversarial attack and defense technologies in natural language processing: A survey. *Neurocomputing*, 492:278–307, 2022.

[Ren *et al.*, 2019] 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*, pages 1085–1097, 2019.

[Sanh *et al.*, 2019] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[Schuster and Paliwal, 1997] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 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, 2013.

[Stahlberg, 2020] Felix Stahlberg. Neural machine translation: A review. *Journal of Artificial Intelligence Research*, 69:343–418, 2020.

[Szegedy *et al.*, 2013] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[Wang *et al.*, 2022] Wenxuan Wang, Wenxiang Jiao, Yongchang Hao, Xing Wang, Shuming Shi, Zhaopeng Tu, and Michael Lyu. Understanding and improving sequence-to-sequence pretraining for neural machine translation. *arXiv preprint arXiv:2203.08442*, 2022.

[Wang *et al.*, 2023] Wenxuan Wang, Jen-tse Huang, Weibin Wu, Jianping Zhang, Yizhan Huang, Shuqing Li, Pinjia He, and Michael Lyu. Mttm: Metamorphic testing for textual content moderation software. *arXiv preprint arXiv:2302.05706*, 2023.

[Williams *et al.*, 2018] 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)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[Zang *et al.*, 2020] 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*, pages 6066–6080, Online, July 2020. Association for Computational Linguistics.

[Zeng *et al.*, 2021] Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 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, 2021.

[Zhang *et al.*, 2022] Jianping Zhang, Weibin Wu, Jen-tse Huang, Yizhan Huang, Wenxuan Wang, Yuxin Su, and Michael R Lyu. Improving adversarial transferability via neuron attribution-based attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14993–15002, 2022.

[Zhang *et al.*, 2023a] Jianping Zhang, Jen-tse Huang, Wenxuan Wang, Yichen Li, Weibin Wu, Xiaosen Wang, Yuxin Su, and Michael R Lyu. Improving the transferability of adversarial samples by path-augmented method. *arXiv preprint arXiv:2303.15735*, 2023.

[Zhang *et al.*, 2023b] Jianping Zhang, Yizhan Huang, Weibin Wu, and Michael R Lyu. Transferable adversarial attacks on vision transformers with token gradient regularization. *arXiv preprint arXiv:2303.15754*, 2023.