BERT-ATTACK: Adversarial Attack Against BERT Using BERT

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, Xipeng Qiu*
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, China
{linyangli19,rtma19,qpguo16,xyxue,xpqiu}@fudan.edu.cn

Abstract

Adversarial attacks for discrete data (such as texts) have been proved significantly more challenging than continuous data (such as images) since it is difficult to generate adversarial samples with gradient-based methods. Current successful attack methods for texts usually adopt heuristic replacement strategies on the character or word level, which remains challenging to find the optimal solution in the massive space of possible combinations of replacements while preserving semantic consistency and language fluency. In this paper, we propose **BERT-Attack**, a high-quality and effective method to generate adversarial samples using pre-trained masked language models exemplified by BERT. We turn BERT against its fine-tuned models and other deep neural models in downstream tasks so that we can successfully mislead the target models to predict incorrectly. Our method outperforms state-of-the-art attack strategies in both success rate and perturb percentage, while the generated adversarial samples are fluent and semantically preserved. Also, the cost of calculation is low, thus possible for large-scale generations. The code is available at https://github.com/LinyangLee/BERT-Attack.

1 Introduction

Despite the success of deep learning, recent works have found that these neural networks are vulnerable to adversarial samples, which are crafted with small perturbations to the original inputs (Goodfellow et al., 2014; Kurakin et al., 2016; Chakraborty et al., 2018). That is, these adversarial samples are imperceptible to human judges while they can mislead the neural networks to incorrect predictions. Therefore, it is essential to explore these adversarial attack methods since the ultimate goal is to make sure the neural networks are highly reliable and robust. While in computer vision fields, both attack strategies and their defense countermeasures are well-explored (Chakraborty et al., 2018), the adversarial attack for text is still challenging due to the discrete nature of languages. Generating of adversarial samples for texts needs to possess such qualities: (1) imperceptible to human judges yet misleading to neural models; (2) fluent in grammar and semantically consistent with original inputs.

Previous methods craft adversarial samples mainly based on specific rules (Li et al., 2018; Gao et al., 2018; Yang et al., 2018; Alzantot et al., 2018; Ren et al., 2019; Jin et al., 2019; Zang et al., 2020). Therefore, these methods are difficult to guarantee the fluency and semantically preservation in the generated adversarial samples at the same time. Plus, these manual craft methods are rather complicated. They use multiple linguistic constraints like NER tagging or POS tagging. Introducing contextualized language models to serve as an automatic perturbation generator could make these rules designing much easier.

The recent rise of pre-trained language models, such as BERT (Devlin et al., 2018), push the performances of NLP tasks to a new level. On the one hand, the powerful ability of a fine-tuned BERT on downstream tasks makes it more challenging to be adversarial attacked (Jin et al., 2019). On the other hand, BERT is a pre-trained masked language model on extremely large-scale unsupervised data and has learned general-purpose language knowledge. Therefore, BERT has the potential to generate more fluent and semantic-consistent substitutions for an input text. Naturally, both the properties of BERT motivate us to explore the possibility of attacking a fine-tuned BERT with another BERT as the attacker.

In this paper, we propose an effective and high-quality adversarial sample generation method: **BERT-Attack**, using BERT as a language model

*Corresponding author.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6193–6202,
November 16–20, 2020.

to generate adversarial samples. The core algorithm of BERT-Attack is straightforward and consists of two stages: finding the vulnerable words in one given input sequence for the target model; then applying BERT in a semantic-preserving way to generate substitutes for the vulnerable words. With the ability of BERT, the perturbations are generated considering the context around. Therefore, the perturbations are fluent and reasonable. We use the masked language model as a perturbation generator and find perturbations that maximize the risk of making wrong predictions (Goodfellow et al., 2014). Differently from previous attacking strategies that require traditional single-direction language models as a constraint, we only need to inference the language model once as a perturbation generator rather than repeatedly using language models to score the generated adversarial samples in a trial and error process.

Experimental results show that the proposed BERT-Attack method successfully fooled its fine-tuned downstream model with the highest attack success rate compared with previous methods. Meanwhile, the perturb percentage and the query number are considerably lower, while the semantic preservation is high.

To summarize our main contributions:

- We propose a simple and effective method, named **BERT-Attack**, to effectively generate fluent and semantically-preserved adversarial samples that can successfully mislead state-of-the-art models in NLP, such as fine-tuned BERT for various downstream tasks.
- BERT-Attack has a higher attacking success rate and a lower perturb percentage with fewer access numbers to the target model compared with previous attacking algorithms, while does not require extra scoring models therefore extremely effective.

2 Related Work

To explore the robustness of neural networks, adversarial attacks have been extensively studied for continuous data (such as images) (Goodfellow et al., 2014; Nguyen et al., 2015; Chakraborty et al., 2018). The key idea is to find a minimal perturbation that maximizes the risk of making wrong predictions. This minimax problem can be easily achieved by applying gradient descent over the continuous space of images (Miyato et al., 2017). However, adversarial attack for discrete data such as text remains challenging.

Adversarial Attack for Text

Current successful attacks for text usually adopt heuristic rules to modify the characters of a word (Jin et al., 2019), and substituting words with synonyms (Ren et al., 2019). Li et al. (2018); Gao et al. (2018) apply perturbations based on word embeddings such as Glove (Pennington et al., 2014), which is not strictly semantically and grammatically coordinated. Alzantot et al. (2018) adopts language models to score the perturbations generated by searching for close meaning words in the word embedding space (Mrkšić et al., 2016), using a trial and error process to find possible perturbations, yet the perturbations generated are still not context-aware and heavily rely on cosine similarity measurement of word embeddings. Glove embeddings do not guarantee similar vector space with cosine similarity distance, therefore the perturbations are less semantically consistent. Jin et al. (2019) apply a semantically enhanced embedding (Mrkšić et al., 2016), which is context unaware, thus less consistent with the unperturbed inputs. Liang et al. (2017) use phrase-level insertion and deletion, which produces unnatural sentences inconsistent with the original inputs, lacking fluency control. To preserve semantic information, Glockner et al. (2018) replace words manually to break the language inference system (Bowman et al., 2015). Jia and Liang (2017) propose manual craft methods to attack machine reading comprehension systems. Lei et al. (2019) introduce replacement strategies using embedding transition.

Although the above approaches have achieved good results, there is still much room for improvement regarding the perturbed percentage, attacking success rate, grammatical correctness and semantic consistency, etc. Moreover, the substitution strategies of these approaches are usually non-trivial, resulting in that they are limited to specific tasks.

Adversarial Attack against BERT

Pre-trained language models have become mainstream for many NLP tasks. Works such as (Wallace et al., 2019; Jin et al., 2019; Pruthi et al., 2019) have explored these pre-trained language models from many different angles. Wallace et al. (2019) explored the possible ethical problems of learned knowledge in pre-trained models.

3 BERT-Attack

Motivated by the interesting idea of turning BERT against BERT, we propose **BERT-Attack**, using the original BERT model to craft adversarial samples to fool the fine-tuned BERT model.

Our method consists of two steps: (1) finding the vulnerable words for the target model and then (2) replacing them with the semantically similar and grammatically correct words until a successful attack.

The most-vulnerable words are the keywords that help the target model make judgments. Perturbations over these words can be most beneficial in crafting adversarial samples. After finding which words that we are aimed to replace, we use masked language models to generate perturbations based on the top-K predictions from the masked language model.

3.1 Finding Vulnerable Words

Under the black-box scenario, the logit output by the target model (fine-tuned BERT or other neural models) is the only supervision we can get. We first select the words in the sequence which have a high significance influence on the final output logit.

Let $S = [w_0, \cdots, w_i \cdots]$ denote the input sentence, and $o_y(S)$ denote the logit output by the target model for correct label y, the importance score I_{w_i} is defined as

$$I_{w_i} = o_y(S) - o_y(S_{\backslash w_i}), \tag{1}$$

where $S_{\backslash w_i} = [w_0, \cdots, w_{i-1}, [\texttt{MASK}], w_{i+1}, \cdots]$ is the sentence after replacing w_i with $[\texttt{MASK}]$.

Then we rank all the words according to the ranking score I_{w_i} in descending order to create word list L. We only take ϵ percent of the most important words since we tend to keep perturbations minimum.

This process maximizes the risk of making wrong predictions, which is previously done by calculating gradients in image domains. The problem is then formulated as replacing these most vulnerable words with semantically consistent perturbations.

3.2 Word Replacement via BERT

After finding the vulnerable words, we iteratively replace the words in list L one by one to find perturbations that can mislead the target model. Previous approaches usually use multiple human-crafted rules to ensure the generated example is semantically consistent with the original one and grammatically correct, such as a synonym dictionary (Ren et al., 2019), POS checker (Jin et al., 2019), semantic similarity checker (Jin et al., 2019), etc. Alzantot et al. (2018) applies a traditional language model to score the perturbed sentence at every attempt of replacing a word.

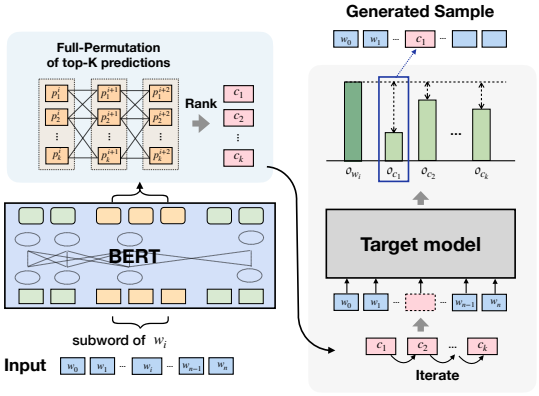

Figure 1: One step of our replacement strategy.

These strategies of generating substitutes are unaware of the context between the substitution positions (usually using language models to test the substitutions), thus are insufficient in fluency control and semantic consistency. More importantly, using language models or POS checkers in scoring the perturbed samples is costly since this trial and error process requires massive inference time.

To overcome the lack of fluency control and semantic preservation by using synonyms or similar words in the embedding space, we leverage BERT for word replacement. The genuine nature of the masked language model makes sure that the generated sentences are relatively fluent and grammar-correct, also preserve most semantic information, which is later confirmed by human evaluators. Further, compared with previous approaches using rule-based perturbation strategies, the masked language model prediction is context-aware, thus dynamically searches for perturbations rather than simple synonyms replacing.

Different from previous methods using complicated strategies to score and constrain the perturbations, the contextualized perturbation generator generates minimal perturbations with only one forward pass. Without running additional neural models to score the sentence, the time-consuming part is accessing the target model only. Therefore the process is extremely efficient.

Algorithm 1 BERT-Attack

procedure WORD IMPORTANCE RANKING
  $S = [w_0, w_1, \cdots]$ // input: tokenized sentence
  $Y \leftarrow$ gold-label
  for w_i in S **do**
    calculate importance score I_{w_i} using Eq. 1
  select word list $L = [w_{top-1}, w_{top-2}, \cdots]$
  // sort S using I_{w_i} in descending order and collect $top - K$ words
procedure REPLACEMENT USING BERT
  $H = [h_0, \cdots, h_n]$ // sub-word tokenized sequence of S
  generate top-K candidates for all sub-words using BERT and get $P^{\in n \times K}$
  for w_j in L **do**
    if w_j is a whole word **then**
      get candidate $C = Filter(P^j)$
      replace word w_j
    else
      get candidate C using PPL ranking and Filter
      replace sub-words $[h_j, \cdots, h_{j+t}]$
    Find Possible Adversarial Sample
    for c_k in C **do**
      $S^{'} = [w_0, \cdots, w_{j-1}, c_k, \cdots]$ // attempt
      if $\text{argmax}(o_y(S^{'}))! = Y$ **then**
        return $S^{adv} = S^{'}$ // success attack
      else
        if $o_y(S^{'}) < o_y(S^{adv})$ **then**
          $S^{adv} = [w_0, \cdots, w_{j-1}, c, \cdots]$ // do one perturbation
  return **None**

Thus, using the masked language model as a contextualized perturbation generator can be one possible solution to craft high-quality adversarial samples efficiently.

3.2.1 Word Replacement Strategy

As seen in Figure 1, given a chosen word w to be replaced, we apply BERT to predict the possible words that are similar to w yet can mislead the target model. Instead of following the masked language model settings, we do not mask the chosen word w and use the original sequence as input, which can generate more semantic-consistent substitutes (Zhou et al., 2019). For instance, given a sequence *"I like the cat."*, if we mask the word *cat*, it would be very hard for a masked language model to predict the original word *cat* since it could be just as fluent if the sequence is *"I like the dog."*. Further, if we mask out the given word w, for each iteration we would have to rerun the masked language model prediction process which is costly.

Since BERT uses Bytes-Pair-Encoding (BPE) to tokenize the sequence $S = [w_0, \cdots, w_i, \cdots]$ into sub-word tokens: $H = [h_0, h_1, h_2, \cdots]$, we need to align the chosen word to its corresponding sub-words in BERT.

Let $\mathcal{M}$ denote the BERT model, we feed the tokenized sequence H into the BERT $\mathcal{M}$ to get output prediction $P = \mathcal{M}(H)$. Instead of using the argmax prediction, we take the most possible K predictions at each position, where K is a hyper-parameter.

We iterate words that are sorted by word importance ranking process to find perturbations. The BERT model uses BPE encoding to construct vocabularies. While most words are still single words, rare words are tokenized into sub-words. Therefore, we treat single words and sub-words separately to generate the substitutes.

Single words For a single word w_j, we make attempts using the corresponding top-K prediction candidates P^j. We first filter out stop words collected from NLTK; for sentiment classifica-

tion tasks we filter out antonyms using synonym dictionaries (Mrkšić et al., 2016) since BERT masked language model does not distinguish synonyms and antonyms. Then for given candidate c_k we construct a perturbed sequence $H^{'} = [h_0, \cdots, h_{j-1}, c_k, h_{j+1} \cdots]$. If the target model is already fooled to predict incorrectly, we break the loop to obtain the final adversarial sample H^{adv}; otherwise, we select from the filtered candidates to pick one best perturbation and turn to the next word in word list L.

Sub-words For a word that is tokenized into sub-words in BERT, we cannot obtain its substitutes directly. Thus we use the perplexity of sub-word combinations to find suitable word substitutes from predictions in the sub-word level. Given sub-words $[h_0, h_1, \cdots, h_t]$ of word w, we list all possible combinations from the prediction $P^{\in t \times K}$ from $\mathcal{M}$, which is K^t sub-word combinations, we can convert them back to normal words by reversing the BERT tokenization process. We feed these combinations into the BERT-MLM to get the perplexity of these combinations. Then we rank the perplexity of all combinations to get the top-K combinations to find the suitable sub-word combinations.

Given the suitable perturbations, we replace the original word with the most likely perturbation and repeat this process by iterating the importance word ranking list to find the final adversarial sample. In this way, we acquire the adversarial samples S^{adv} effectively since we only iterate the masked language model once and do perturbations using the masked language model without other checking strategies.

We summarize the two-step BERT-Attack process in Algorithm 1.

4 Experiments

4.1 Datasets

We apply our method to attack different types of NLP tasks in the form of text classification and natural language inference. Following Jin et al. (2019), we evaluate our method on 1k test samples randomly selected from the test set of the given task which are the same splits used by Alzantot et al. (2018); Jin et al. (2019). The GA method only uses a subset of 50 samples in the FAKE, IMDB dataset.

Text Classification We use different types of text classification tasks to study the effectiveness of our method.

- **Yelp** Review classification dataset, containing. Following Zhang et al. (2015), we process the dataset to construct a polarity classification task.
- **IMDB** Document-level movie review dataset, where the average sequence length is longer than the Yelp dataset. We process the dataset into a polarity classification task [1].
- **AG's News** Sentence level news-type classification dataset, containing 4 types of news: World, Sports, Business, and Science.
- **FAKE** Fake News Classification dataset, detecting whether a news document is fake from Kaggle Fake News Challenge [2].

Natural Language Inference

- **SNLI** Stanford language inference task (Bowman et al., 2015). Given one premise and one hypothesis, and the goal is to predict if the hypothesis is entailment, neural, or contradiction of the premise.
- **MNLI** Language inference dataset on multi-genre texts, covering transcribed speech, popular fiction, and government reports (Williams et al., 2018), which is more complicated with diversified written and spoken style texts, compared with the SNLI dataset, including eval data matched with training domains and eval data mismatched with training domains.

4.2 Automatic Evaluation Metrics

To measure the quality of the generated samples, we set up various automatic evaluation metrics. The success rate, which is the counter-part of after-attack accuracy, is the core metric measuring the success of the attacking method. Meanwhile, the perturbed percentage is also crucial since, generally, less perturbation results in more semantic consistency. Further, under the black-box setting, queries of the target model are the only accessible information. Constant queries for one sample is less applicable. Thus query number per sample is also a key metric. As used in TextFooler (Jin et al., 2019), we also use Universal Sentence Encoder (Cer et al., 2018) to measure the semantic consistency between the adversarial sample and the original sequence. To balance between semantic preservation and attack success rate, we set up a threshold of semantic similarity score to filter the less similar examples.

[1] https://datasets.imdbws.com/

[2] https://www.kaggle.com/c/fake-news/data

Dataset	Method	Original Acc	Attacked Acc	Perturb %	Query Number	Avg Len	Semantic Sim
Fake	BERT-Attack(ours)	97.8	**15.5**	**1.1**	**1558**	885	**0.81**
	TextFooler(Jin et al., 2019)		19.3	11.7	4403		0.76
	GA(Alzantot et al., 2018)		58.3	1.1	28508		-
Yelp	BERT-Attack(ours)	95.6	**5.1**	**4.1**	**273**	157	**0.77**
	TextFooler		6.6	12.8	743		0.74
	GA		31.0	10.1	6137		-
IMDB	BERT-Attack(ours)	90.9	**11.4**	**4.4**	**454**	215	**0.86**
	TextFooler		13.6	6.1	1134		**0.86**
	GA		45.7	4.9	6493		-
AG	BERT-Attack(ours)	94.2	**10.6**	**15.4**	**213**	43	**0.63**
	TextFooler		12.5	22.0	357		0.57
	GA		51	16.9	3495		-
SNLI	BERT-Attack(ours)	89.4(H/P)	7.4/**16.1**	**12.4/9.3**	**16/30**	8/18	0.40/**0.55**
	TextFooler		**4.0**/20.8	18.5/33.4	60/142		**0.45**/0.54
	GA		14.7/-	20.8/-	613/-		-
MNLI matched	BERT-Attack(ours)	85.1(H/P)	**7.9/11.9**	**8.8/7.9**	**19/44**	11/21	0.55/**0.68**
	TextFooler		9.6/25.3	15.2/26.5	78/152		**0.57**/0.65
	GA		21.8/-	18.2/-	692/-		-
MNLI mismatched	BERT-Attack(ours)	82.1(H/P)	**7/13.7**	**8.0/7.1**	**24/43**	12/22	0.53/**0.69**
	TextFooler		8.3/22.9	14.6/24.7	86/162		**0.58**/0.65
	GA		20.9/-	19.0/-	737/-		-

Table 1: Results of attacking against various fine-tuned BERT models. TextFooler is the state-of-the-art baseline. For MNLI task, we attack the hypothesis(H) or premises(P) separately.

4.3 Attacking Results

As shown in Table 1, the BERT-Attack method successfully fool its downstream fine-tuned model. In both text classification and natural language inference tasks, the fine-tuned BERTs fail to classify the generated adversarial samples correctly.

The average after-attack accuracy is lower than 10%, indicating that most samples are successfully perturbed to fool the state-of-the-art classification models. Meanwhile, the perturb percentage is less than 10 %, which is significantly less than previous works.

Further, **BERT-Attack** successfully attacked all tasks listed, which are in diversified domains such as News classification, review classification, language inference in different domains. The results indicate that the attacking method is robust in different tasks. Compared with the strong baseline introduced by Jin et al. (2019)[3] and Alzantot et al. (2018)[4], the BERT-Attack method is more efficient and more imperceptible. The query number and the perturbation percentage of our method are much less.

We can observe that it is generally easier to attack the review classification task since the perturb percentage is incredibly low. BERT-Attack can mislead the target model by replacing a handful of words only. Since the average sequence length is relatively long, the target model tends to make judgments by only a few words in a sequence, which is not the natural way of human prediction. Thus, the perturbation of these keywords would result in incorrect prediction from the target model, revealing the vulnerability of it.

4.4 Human Evaluations

For further evaluation of the generated adversarial samples, we set up human evaluations to measure the quality of the generated samples in fluency and grammar as well as semantic preservation.

We ask human judges to score the grammar correctness of the mixed sentences of generated ad-

[3]https://github.com/jind11/TextFooler
[4]https://github.com/QData/TextAttack

versarial samples and original sequences, scoring from 1-5 following Jin et al. (2019). Then we ask human judges to make predictions in a shuffled mix of original and adversarial texts. We use the IMDB dataset and the MNLI dataset, and for each task, we select 100 samples of both original and adversarial samples for human judges. We ask three human annotators to evaluate the examples. For label prediction, we take the majority class as the predicted label, and for semantic and grammar check we use an average score among the annotators.

Seen in Table 2, the semantic score and the grammar score of the adversarial samples are close to the original ones. MNLI task is a sentence pair prediction task constructed by human crafted hypotheses based on the premises, therefore original pairs share a considerable amount of same words. Perturbations on these words would make it difficult for human judges to predict correctly therefore the accuracy is lower than simple sentence classification tasks.

Dataset		Accuracy	Semantic	Grammar
MNLI	Original	0.90	3.9	4.0
	Adversarial	0.70	3.7	3.6
IMDB	Original	0.91	4.1	3.9
	Adversarial	0.85	3.9	3.7

Table 2: Human-Evaluation Results.

4.5 BERT-Attack against Other Models

The BERT-Attack method is also applicable in attacking other target models, not limited to its fine-tuned model only. As seen in Table 3, the attack is successful against LSTM-based models, indicating that BERT-Attack is feasible for a wide range of models. Under BERT-Attack, the ESIM model is more robust in the MNLI dataset. We assume that encoding two sentences separately gets higher robustness. In attacking BERT-large models, the performance is also excellent, indicating that BERT-Attack is successful in attacking different pre-trained models not only against its own fine-tuned downstream models.

5 Ablations and Discussions

5.1 Importance of Candidate Numbers

The candidate pool range is the major hyper-parameter used in the BERT-Attack algorithm. As seen in Figure 2, the attack rate is rising along with the candidate size increasing. Intuitively, a larger

Dataset	Model	Ori Acc	Atk Acc	Perturb %
IMDB	Word-LSTM	89.8	10.2	2.7
	BERT-Large	98.2	12.4	2.9
Yelp	Word-LSTM	96.0	1.1	4.7
	BERT-Large	97.9	8.2	4.1
MNLI matched	ESIM	76.2	9.6	21.7
	BERT-Large	86.4	13.2	7.4

Table 3: BERT-Attack against other models.

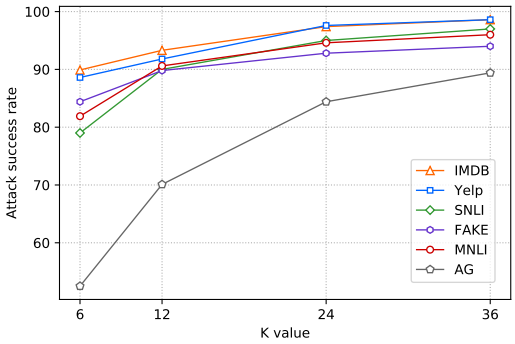

Figure 2: Using different candidate number K in the attacking process.

K would result in less semantic similarity. However, the semantic measure via Universal Sentence Encoder is maintained in a stable range, (experiments show that semantic similarities drop less than 2%), indicating that the candidates are all reasonable and semantically consistent with the original sentence.

Further, a fixed candidate number could be rigid in practical usage, so we run a test using a threshold to cut off candidates that are less possible as a plausible perturbation.

As seen in Table 4, when using a flexible threshold to cut off unsuitable candidates, the attacking process has a lower query number. This indicates that some candidates predicted by the masked language model with a lower prediction score may not be meaningful so skipping these candidates can save the unnecessary queries.

Dataset	Method	Ori Acc	Atk Acc	Queries %
IMDB	Fixed-K	90.9	11.4	454
	With Threshold	90.9	12.4	440

Table 4: Flexible Candidates Using a threshold to cut off unsuitable candidates.

5.2 Importance of Sequence Length

The BERT-Attack method is based on the contextualized masked language model. Thus the sequence length plays an important role in the high-quality perturbation process. As seen, instead of the previous methods focusing on attacking the hypothesis of the NLI task, we aim at premises whose average length is longer. This is because we believe that contextual replacement would be less reasonable when dealing with extremely short sequences. To avoid such a problem, we believe that many word-level synonym replacement strategies can be combined with BERT-Attack, allowing the BERT-Attack method to be more applicable.

Dataset	Method	Ori Acc	Atk Acc	Perturb %
MNLI matched	BERT-Atk	85.1	7.9	8.8
	+Adv Train	84.6	23.1	10.5

Table 5: Adversarial training results.

Dataset	Model	LSTM	BERT-base	BERT-large
IMDB	Word-LSTM	-	0.78	0.75
	BERT-base	0.83	-	0.71
	BERT-large	0.87	0.86	-
Dataset	**Model**	**ESIM**	**BERT-base**	**BERT-large**
MNLI	ESIM	-	0.59	0.60
	BERT-base	0.60	-	0.45
	BERT-large	0.59	0.43	-

Table 6: Transferability analysis using attacked accuracy as the evaluation metric. The column is the target model used in attack, and the row is the tested model.

5.3 Transferability and Adversarial Training

To test the transferability of the generated adversarial samples, we take samples aimed at different target models to attack other target models. Here, we use BERT-base as the masked language model for all different target models. As seen in Table 6, samples are transferable in NLI task while less transferable in text classification.

Meanwhile, we further fine-tune the target model using the generated adversarial samples from the train set and then test it on the test set used before. As seen in Table 5, generated samples used in fine-tuning help the target model become more robust while accuracy is close to the model trained with clean datasets. The attack becomes more difficult, indicating that the model is harder to be attacked. Therefore, the generated dataset can be used as additional data for further exploration of making neural models more robust.

Dataset	Model	Atk Acc	Perturb %	Semantic
Yelp	BERT-Atk	5.1	4.1	0.77
	w/o sub-word	7.1	4.3	0.74
MNLI	BERT-Atk	11.9	7.9	0.68
	w/o sub-word	14.7	9.3	0.63

Table 7: Effects on sub-word level attack.

5.4 Effects on Sub-Word Level Attack

BPE method is currently the most efficient way to deal with a large number of words, as used in BERT. We establish a comparative experiment where we do not use the sub-word level attack. That is we skip those words that are tokenized with multiple sub-words.

As seen in Table 7, using the sub-word level attack can achieve higher performances, not only in higher attacking success rate but also in less perturbation percentage.

Dataset	Method	Atk Acc	Perturb %	Semantic
MNLI matched	MIR	7.9	8.8	0.68
	Random	20.2	12.2	0.60
	LIR	27.2	15.0	0.60

Table 8: Most Importance Ranking (MIR) vs Least Importance Ranking (LIR)

5.5 Effects on Word Importance Ranking

Word importance ranking strategy is supposed to find keys that are essential to NN models, which is very much like calculating the maximum risk of wrong predictions in the FGSM algorithm (Goodfellow et al., 2014). When not using word importance ranking, the attacking algorithm is less successful.

Dataset	Method	Runtime(s/sample)
IMDB	BERT-Attack(w/o BPE)	14.2
	BERT-Attack(w/ BPE)	16.0
	Textfooler(Jin et al., 2019)	42.4
	GA(Alzantot et al., 2018)	2582.0

Table 9: Runtime comparison.

Dataset					Label
MNLI	Ori	Some rooms have balconies .	Hypothesis	All of the rooms have balconies off of them .	Contradiction
	Adv	Many rooms have balconies .	Hypothesis	All of the rooms have balconies off of them .	Neutral
IMDB	Ori	it is hard for a lover of the novel northanger abbey to sit through this bbc adaptation and to keep from throwing objects at the tv screen... why are so many facts concerning the tilney family and mrs . tilney ' s death altered unnecessarily ? to make the story more ' horrible ? '			Negative
	Adv	it is hard for a lover of the novel northanger abbey to sit through this bbc adaptation and to keep from throwing objects at the tv screen... why are so many facts concerning the tilney family and mrs . tilney ' s death altered unnecessarily ? to make the plot more ' horrible ? '			Positive
IMDB	Ori	i first seen this movie in the early 80s .. it really had nice picture quality too . anyways , i 'm glad i found this movie again ... the part i loved best was when he hijacked the car from this poor guy... this is a movie i could watch over and over again . i highly recommend it .			Positive
	Adv	i first seen this movie in the early 80s .. it really had nice picture quality too . anyways , i 'm glad i found this movie again ... the part i loved best was when he hijacked the car from this poor guy... this is a movie i could watch over and over again . i inordinately recommend it .			Negative

Table 10: Some generated adversarial samples. Origin label is the correct prediction while label is adverse prediction. Only red color parts are perturbed. We only attack premises in MNLI task. Text in FAKE dataset and IMDB dataset is cut to fit in the table. Original text contains more than 200 words.

5.6 Runtime Comparison

Since BERT-Attack does not use language models or sentence encoders to measure the output sequence during the generation process, also, the query number is lower, therefore the runtime is faster than previous methods. As seen in Table 9, BERT-Attack is much faster than generic algorithm (Alzantot et al., 2018) and 3 times faster then Textfooler.

5.7 Examples of Generated Adversarial Sentences

As seen in Table 10, the generated adversarial samples are semantically consistent with its original input, while the target model makes incorrect predictions. In both review classification samples and language inference samples, the perturbations do not mislead human judges.

6 Conclusion

In this work, we propose a high-quality and effective method **BERT-Attack** to generate adversarial samples using BERT masked language model. Experiment results show that the proposed method achieves a high success rate while maintaining a minimum perturbation. Nevertheless, candidates generated from the masked language model can sometimes be antonyms or irrelevant to the original words, causing a semantic loss. Thus, enhancing language models to generate more semantically related perturbations can be one possible solution to perfect BERT-Attack in the future.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. We are thankful for the help of Demin Song, Hang Yan and Pengfei Liu. This work was supported by the National Natural Science Foundation of China (No. 61751201, 62022027 and 61976056), Shanghai Municipal Science and Technology Major Project (No. 2018SHZDZX01) and ZJLab.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *CoRR*, abs/1804.07998.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

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

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. *arXiv preprint arXiv:1805.02266*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is BERT really robust? natural language attack on text classification and entailment. *CoRR*, abs/1907.11932.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Qi Lei, Lingfei Wu, Pin-Yu Chen, Alexandros G Dimakis, Inderjit S Dhillon, and Michael Witbrock. 2019. Discrete adversarial attacks and submodular optimization with applications to text classification. *Systems and Machine Learning (SysML)*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.

Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. 2017. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. volume 41, pages 1979–1993.

Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1532–1543.

Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. 2019. Combating adversarial misspellings with robust word recognition. *arXiv preprint arXiv:1905.11268*.

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.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. *Empirical Methods in Natural Language Processing*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1112–1122.

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I Jordan. 2018. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *arXiv preprint arXiv:1805.12316*.

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.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy. Association for Computational Linguistics.