A Faster Adversarial Attack Method on LLMs Bringing More Transferable Samples

Anonymous ACL submission

Abstract

With the surge in research on LLMs, various methods for evaluating and attacking the robustness of LLMs have emerged and attracted increasing attention. Traditional adversarial attack methods often have a high dependency on the victim model, leading to poor transferability of generated adversarial samples. The applicability of the obtained attack samples is limited to the current white-box model, making it difficult to transfer attacks to other black-box models. In the scenario of LLMs, problems like poor attack effectiveness and slow attack speed become more pronounced in traditional adversarial attack methods. Through the analysis of traditional text adversarial attack methods, we propose a method capable of producing attack samples with better transferability. Additionally, it enhances attack success rates and greatly improves attack speed.

1 Introduction

Currently, large language models (LLMs) like LLaMA, ChatGPT (Touvron et al., 2023; Ouyang et al., 2022) have shown significant potential in different downstream tasks (Kasneci et al., 2023; Thirunavukarasu et al., 2023; Liu et al., 2023). However, a series of studies on adversarial attacks (Li et al., 2020b; Liu et al., 2020) have found that minor perturbations based on the original input can greatly affect the performance of PLMs. On existing text adversarial attack datasets (Zhu et al., 2023), the performance of various LLMs is quite impressive. This indicates that traditional adversarial attack methods struggle to overcome LLMs, highlighting the increasing importance of developing an efficient attack method that can effectively overcome LLMs.

Despite the success of adversarial attacks in image and speech domains (Chakraborty et al., 2018; Kurakin et al., 2018; Carlini and Wagner, 2018), limitations persist in Natural Language Processing (NLP) due to the discrete nature of language (Studdert-Kennedy, 2005; Armstrong et al., 1995). The primary challenge is to identify a suitable search algorithm so as to effectively perturb the victim model and mislead its judgments (Morris et al., 2020; Yoo and Qi, 2021). Recent research (Jin et al., 2020) emphasizes preserving specific characteristics:

	CNN	LSTM	Bert	LLaMA	Baichuan
CNN	**1.6**	79.4	82.5	81.5	81.6
LSTM	85.9	**5.3**	84.3	83.3	83.4
Bert	90.8	89.7	**10.5**	86.4	86.8
LLaMA	91.2	93.6	94.8	**16.4**	88.4
Baichuan	87.4	83.5	87.8	81.5	**13.7**
ChatGPT	90.1	88.4	89.7	**85.5**	85.7
Average	89.1	86.9	87.8	**83.6**	85.2

Table 1: Transferability evaluation of Bert-Attack Samples on IMDB dataset. Row i and column j is the Accuracy of samples generated from model j and evaluated on model i. The Average result is from non-diagonal elements of each column

1) Consistency with human predictions, maintaining human judgment while misleading Language Models;

2) Semantic similarity, retaining the semantics of the original input;

3) Language fluency, ensuring grammatical correctness.

Mainstream approaches often use a two-step process: first, ranking token importance based on the original input; then, sequentially replacing these tokens using specific rules. However, current methods face two limitations. On one hand, they use the same Victim model for both individual word importance assessment and subsequent replacement attacks. This makes the Victim model serve as both a reference for information and a target for attack, making it relatively easy to be compromised and posing a risk to the poor transferability of generated adversarial samples which can be seen at Table 1. On the other hand, their determined editing replacement sequence proceeds one by one in a sequential manner. The initial few replacements will not successfully attacking the victim model, yet each step requires model inference, leading to unnecessary time overhead.

To tackle the mentioned issues, we initially analyze the reasons for the slow speed and poor effectiveness of traditional text adversarial attack methods on LLMs. We removed the Important Score module, which was the most problematic, and used the Ensemble Prompt within ChatGPT to generate a priority queue for Important Level. Based on the Important Level priority queue, we can perform parallel replacement of entire words within the same priority queue, instead of the traditional method of replacing them one by one, as shown in Figure 3. This reduces the times of model inferences, resulting in a significant speed improvement. We also employed the Multi-disturb and Dynamic-disturb approaches to further enhance the attack effectiveness and transferability of generated adversarial samples. The former involves three levels of disturbances within the same sentence, while the latter dynamically adjusts the proportions and thresholds of the three types of disturbances based on the sentence length of input. Both of them significantly increased the attack effectiveness and transferability of attack samples.

The primary contributions of this work can be summarized as follows:

- To our knowledge, we are the first to analyze the reasons behind the slow speed and poor effectiveness of adversarial attacks on LLMs and replace the problematic Important Score with Important Level.

- Our method employs ChatGPT for word classification, fundamentally altering the attack process and sequence, enhancing effectiveness, significantly improving speed, and making defense more difficult.

- Our method enhanced the transferability of adversarial samples which broaden the application scenarios of adversarial attacks and worked well on ChatGPT.

2 Related Work

2.1 Preliminaries: Adversarial Attack

Following Fang et al., 2023, adversarial attacks aim at generating perturbations on inputs which can mislead the models output. These perturbations can be very small, and imperceptible to human senses. Attacks can be targeted, seeking to change the output of the model to a specific class or text string, or untargeted, seeking only to result in an erroneous classification or generation.

As for NLP tasks, given a corpus of N input texts, $\mathbb{X} = \{x_1, x_2, x_3, \ldots, x_N\}$, and an output space $\mathbb{Y} = \{y_1, y_2, y_3, \ldots, y_N\}$ containing K labels, the language model F learns a mapping $f : x \rightarrow y$, which learns to classify each input sample $x \in X$ to the ground-truth label $y_{\text{gold}} \in \mathcal{Y}$:

$$F(x) = \arg\max_{y_i \in \mathcal{Y}} P(y_i|x) \tag{1}$$

The adversary of text $x \in X$ can be formulated as $x_{\text{adv}} = x + \epsilon$, where ϵ is a perturbation to the input x. The goal is to mislead the victim model F within a certain constraint $C(x_{\text{adv}})$:

$$\begin{aligned} &F(x_{\text{adv}}) = \arg\max_{y_i \in \mathcal{Y}} P(y_i|x_{\text{adv}}) \neq F(x), \\ &\text{and } C(x_{\text{adv}}, x) \leq \lambda \end{aligned} \tag{2}$$

where λ is the coefficient, and C(x_{adv}, x) is usually calculated by the semantic or syntactic similarity (Cer et al., 2018; Oliva et al., 2011) between the input x and its corresponding adversary x_{adv}.

Recently, the adversarial attack task has been framed as a combinatorial optimization problem. However, previous studies (Gao et al., 2018; Ren et al., 2019; Yoo and Qi, 2021) address this problem without considering the subsequent influence of substitution at each step, making attack far from the most effective.

2.2 Text Adversarial Attack

For NLP tasks, the adversarial attacks occur at various text levels including the character, the word, or the sentence level. Character-level attacks involve altering text by changing letters, symbols, and numbers. Word-level attacks (Wei and Zou, 2019) involve modifying the vocabulary with synonyms, misspellings, or specific keywords. Sentence-level attacks (Coulombe, 2018; Xie et al., 2020) involve adding crafted sentences to disrupt the model's outcomes.

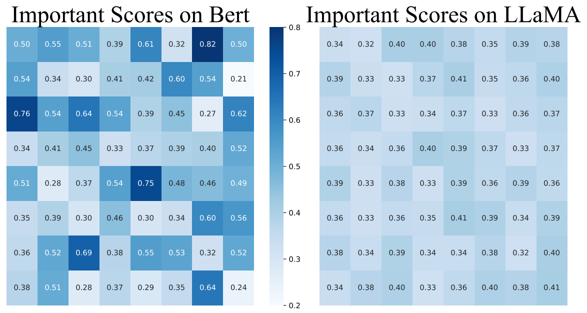

Figure 1: Different Important Scores of the same sentence from Bert-Attack on SA-LLaMA and Bert.

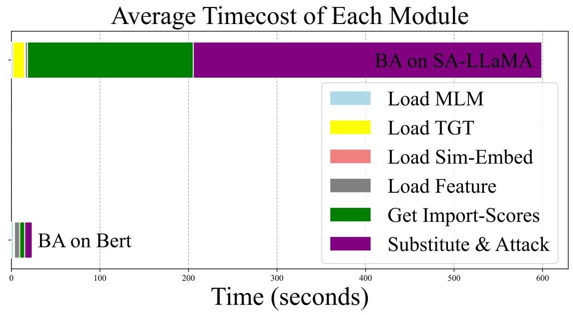

Figure 2: Different time cost of each module from Bert-Attack on SA-LLaMA and Bert.

Current adversarial attacks use substitution tactics to create adversarial examples for NLP tasks(Alzantot et al., 2018; Jin et al., 2020). Most works are focused on score-based black-box attacks, where the output's prediction is available. Diverse strategies, such as genetic algorithms (Zang et al., 2019; Guo et al., 2021), greedy search (Sato et al., 2018; Yoo and Qi, 2021), or gradient-based methods (Ebrahimi et al., 2018; Cheng et al., 2018), are employed to identify substitution words form synonyms (Kuleshov et al., 2018; Jin et al., 2020) or language models (Li et al., 2020b; Garg and Ramakrishnan, 2020; Li et al., 2020a). Some papers have optimized the sampling methods, but they still tend to be time-intensive. Fang et al., 2023 apply reinforcement learning, showing promise on small models but facing challenges on LLMs due to lengthy iterations, limiting large-scale adversarial samples. More details can be found in Appendix A.

3 Method

3.1 Problems of Important Scores

As shown in Figure 1, the same sentence calculated for Important Scores on Bert and LLaMA exhibits a significant difference. The former has a sharper distribution, while the latter essentially lacks substantial numerical differences. The calculation method of Important Scores only involves whether the labels change and the confidence change of the model for the input after masking. On the other hand, Important Scores determine the subsequent attack sequence, which is crucial for the success of subsequent attacks and the number of attacks. In other words, Important Scores are highly correlated with the model architecture. In the subsequent process of traditional attack methods, the confidence assessment of the model remains the most important even **the only** indicator. This results in the success of attacks and the choices within the attack search space being highly related to the architecture of the Victim model, contributing to the poor transferability of generated attack samples, as indicated in Figure 1.

As depicted in Figure 2, when applying Bert-Attack to attack small models like Bert, the average time spent per entry is very short, and the time consumption of various components is not significantly different. However, the time cost per inference on LLMs far exceeds that of Bert, disrupting this balance. We can see that on LLMs, more than 80% of the time spent per entry in the attack is consumed by the get_import_scores component. It's worth noting that this phenomenon is even more pronounced in successfully attacked samples. In this step, it is necessary to mask each sub_word in the statement, and then calculate the importance score during inference based on the changes in model determination confidence and output labels. In other words, the number of inferences required for each statement is the number of sub_words when calculating import_scores plus the subsequent attempts at replacement.

3.2 Important Level

To address the issues caused by Important Scores, we introduce Important Level as a replacement. Specifically, under the instruction of the Ensemble Prompt, ChatGPT partitions all whole words in the original input according to semantic importance. By introducing ChatGPT as a third-party source of partitioning, it can avoid excessive dependence on the Victim model during the attack process. It can also utilize the rich semantic knowledge within ChatGPT, making the subsequent generation of attack sequences more universally semantic.

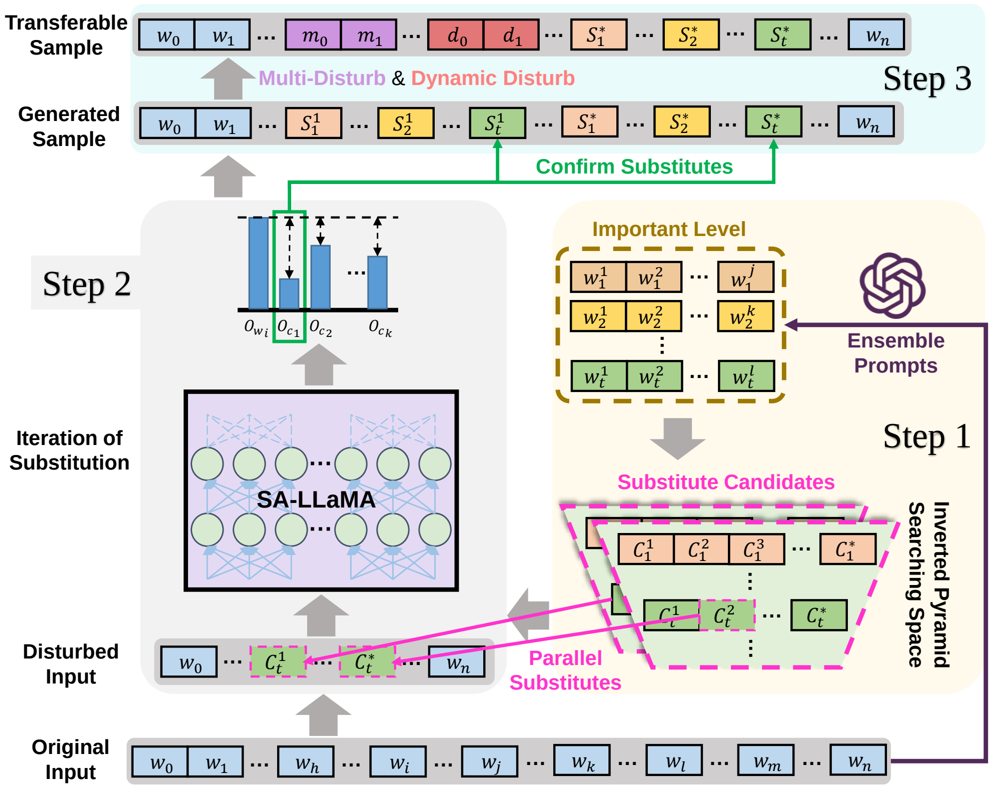
Transferable Sample
w_0 w_1 ··· m_0 m_1 ··· d_0 d_1 ··· S_1^* ··· S_2^* ··· S_t^* ··· w_n
Multi-Disturb & Dynamic Disturb
Step 3
Generated Sample
w_0 w_1 ··· S_1^1 ··· S_2^1 ··· S_t^1 ··· S_1^* ··· S_2^* ··· S_t^* ··· w_n
Confirm Substitutes
Step 2
o_{w_i} o_{c_1} o_{c_2} o_{c_k}
Iteration of Substitution
SA-LLaMA
Disturbed Input
w_0 ··· C_t^1 ··· C_t^* ··· w_n
Important Level
w_1^1 w_1^2 ··· w_1^j
w_2^1 w_2^2 ··· w_2^k
w_t^1 w_t^2 ··· w_t^l
Ensemble Prompts
Step 1
Substitute Candidates
C_1^1 C_1^2 C_1^3 ··· C_1^*
C_t^1 C_t^2 ··· C_t^*
Inverted Pyramid Searching Space
Parallel Substitutes
Original Input
w_0 w_1 ··· w_h ··· w_i ··· w_j ··· w_k ··· w_l ··· w_m ··· w_n

Figure 3: **Step 1:** The *Original Input* uses *Ensemble Prompts* with ChatGPT to categorize words into 5 *Important Levels* with varying word counts. The *Inverted Pyramid Searching Space* reflects the decreasing length of *Substitute Candidates* based on decreasing levels. **Step 2:** Selecting words from the same level and generates a *Disturbed Input* through *Parallel Substitutions*. Exploring possible *Disturbed Inputs* via *SA-LLaMA*, choose the result surpassing the threshold as *Generated Sample* from *Confirmed Substitutions*. *Substitution Iterations* will end when meet the finished condition. **Step 3:** Implementing *Multi-Disturb* and *Dynamic Disturb* produces *Transferable Samples*.

Important Level takes the original sentence as input and outputs a priority queue with 5 levels, with the different number of words in each level. This approach does not require inference from the Victim model, alleviating the problem of high inference costs for LLMs. Moreover, as this stage completely breaks free from dependence on the Victim model, the results of Important Level are universal for all different Victim models. In contrast to Importance Scores, which requires recalculation for each Victim model, this ensures the universality of Important Level results. In other words, the creation of Important Level is more pure, merely the original text input and ChatGPT, rather than evaluations of importance within the abstract semantic space of different models. This ensures that the results obtained are more naturally aligned with linguistic essence, rather than being fitted into the semantic space of the Victim model.

This approach has led to a substantial acceleration and simplification of the stage where the replacement order is determined before the attack starts. LLMs only need one inference to obtain a comprehensive Important Level priority queue, while the traditional approach requires inference times proportional to the length of the text. Additionally, in the subsequent step-by-step replacement process of traditional white-box attack methods, the retention of each candidate replacement result depends on the sample with the most significant decrease in model confidence. Although this greedy search approach achieves a high success rate under low modification rates, it also results in the generated attack samples being extremely overfit to the model architecture. This is the exact reason for the very poor transferability of adversarial attack samples in existing attack methods. In this case, white-box attacks are basically effective only for this specific Victim Model. In the current proliferation of LLMs, their generalization and applicability appear very limited.

Building on Important Level, we partitioned all words into different levels, where words within the same level have no specific order. This allows us to parallelize the replacement of candidate words within the same level, facilitating model inference to evaluate the effectiveness of this replacement. The parallel replacement process, compared to the greedy replacement of sequential word searches, significantly reduces the search space and the number of inferences. This represents a substantial improvement in speeding up the process of attacking LLMs. Additionally, on the decreasing levels of Important Level, we implement a *reverse pyramid search space* to optimize the search space, reducing inefficient search costs. Words with higher priority are considered to have a greater impact on sentence sentiment determination. We need a larger search space to find semantically similar words, attempting to replace with candidate words from synonyms that cause a rapid decline in the Victim model's performance. For words with lower priority levels, we use a smaller search space to find synonyms, as changes to these words have little impact on sentence sentiment determination. Excessive searching for these words may result in increased inference costs without necessarily being effective.

3.3 Multi-Disturb & Dynamic-Disturb

Building on the aforementioned method, to further enhance the robustness of adversarial attack samples, we strategically propose two tricks for optimization. These two strategies can be integrated into other traditional text adversarial attack methods, acting more like a post-processing approach, and can greatly enhance the transferability of adversarial attack samples. Specifically, the Multi-Disturb strategy refers to introducing a variety of disturbances within the same sentence. Appendix H outlines 9 ways of disturbance, including character-level, word-level, and sentence-level disturbances, which can greatly enhance the transferability of attack samples. However, how to set the ratios of these three types of disturbances largely determines the quality of the transferability from generated attack samples. Therefore, the following strategy is proposed.

Dynamic-Disturb refers to using an FFN+Softmax network to assess the length and structural distribution of the input sentence, outputting the ratios of these three types of disturbances. In the step of evaluating whether an attack sample is effective, traditional attack methods almost solely rely on the confidence of model output, a practice that undoubtedly promotes overfitting of attack samples to the model architecture. Therefore, in the process of determining the effectiveness of a replacement, we introduce random disturbance to the decrease in model confidence. This may result in the loss of some already successfully attacked samples, but it also prevents the occurrence of the phenomenon where the attack stops after succeeding on this particular Victim model. Traditional methods rely heavily on model confidence, leading to overfitting. To counter this, we introduce random disturbances during effectiveness assessment, reducing model confidence. This might sacrifice past successful attacks but prevents reliance on the success of the victim model. The attacking algorithm is on Appendix B.

4 Experiments

4.1 Experiments Setups

Tasks and Datasets Following (Li et al., 2020b); we evaluate the effectiveness of our method F-ATTACK mainly on classification tasks upon diverse datasets covering news topics (AG's News; Zhang et al., 2015), sentiment analysis at sentence (MR; Pang and Lee, 2005) and document levels (IMDB[1] and Yelp Polarity; Zhang et al., 2015). Appendix A provides the details of the datasets and basic statistics. Following Jin et al. (2020); Alzantot et al. (2018), we attack 1k samples randomly selected from the test set of each task. The statistics of datasets and more details can be found in Appendix C.

Baselines We compare F-ATTACK with recent studies:1) TextFooler (Jin et al., 2020): find important words via probability weighted word saliency and then apply substitution with counter-fitted word embeddings. 2) BERT-Attack (Li et al., 2020b): use mask-predict approach to generate adversaries. Using codes from authors and TextAttack tools (Morris et al., 2020), we implement these baselines. Applying fairness constraints as Morris et al. (2020) in Appendix D.

Implementation Details We employed LLaMA-2-7B as the base model of LLMs. With the same training settings on various datasets, we finally get specific Task-LLaMA models. Among them, the Task-LLaMA fine-tuned on the IMDB training set

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

Dataset	Method	A-rate↑	Mod↓	Sim↑	Dataset	Method	A-rate ↑	Mod ↓	Sim↑
Yelp	TextFooler	78.9	9.1	0.73	**IMDB**	TextFooler	83.3	8.1	0.79
	BERT-Attack	80.5	11.5	0.69		BERT-Attack	84.2	9.6	0.78
	F-ATTACK (Zero-Shot)	81.3	10.3	0.71		F-ATTACK (Zero-Shot)	86.1	8.7	0.81
	F-ATTACK (Few-Shot)	83.7	**9.0**	**0.77**		F-ATTACK (Few-Shot)	86.7	**7.4**	0.76
	+MD	**84.6**	12.3	0.71		+MD	87.1	10.7	0.77
	+MD +DD	84.5	11.9	0.72		+MD +DD	**87.7**	9.9	**0.81**
AG's News	TextFooler	73.2	16.1	0.54	**MR**	TextFooler	81.3	10.5	0.53
	BERT-Attack	76.6	17.3	**0.59**		BERT-Attack	82.8	10.2	0.51
	F-ATTACK (Zero-Shot)	77.1	18.3	0.52		F-ATTACK (Zero-Shot)	**84.6**	**8.9**	**0.58**
	F-ATTACK (Few-Shot)	81.9	**16.1**	0.58		F-ATTACK (Few-Shot)	83.1	12.4	0.44
	+MD	82.8	19.4	0.53		+MD	83.0	13.2	0.40
	+MD +DD	**83.0**	19.1	0.55		+MD +DD	84.0	10.4	0.50

Table 2: Automatic evaluation results of attack success rate (A-rate), modification rate (Mod), and semantic similarity (Sim) on Task-LLaMA. ↑ represents the higher the better and ↓ means the opposite. The best results are **bolded**, and the second-best ones are underlined.

achieved an accuracy of 96.95% on the test set, surpassing XLNET with additional data, which achieved 96.21%. The model achieved 93.63% on another sentiment classification dataset, SST-2, indicating that Task-LLaMA is not overfit to the training data but indeed possesses quite good sentiment analysis performance. This will provide credibility support for our subsequent efforts to improve adversarial attack methods and generate more robust samples based on it.

Automatic Evaluation Metrics Building on prior work (Jin et al., 2020; Morris et al., 2020), we assess the results with following metrics: 1) Attack success rate (A-rate): post-attack model performance decline; 2) Modification rate (Mod): percentage of altered words compared to the original; 3) Semantic similarity (Sim): cosine similarity between original and adversary texts via universal sentence encoder (USE; Cer et al., 2018); and 4) Transferability (Trans): the mean accuracy decreases across three models between adversarial and original samples.

Manual Evaluation Metrics We further manually validate the quality of the adversaries from three challenging properties. 1) Human prediction consistency (Con): how often human judgment aligns with the true label; 2) Language fluency (Flu): measured on a scale of 1 to 5 for sentence coherence (Gagnon-Marchand et al., 2019); and 3) Semantic similarity (Sim$_{\text{hum}}$): gauging consistency between input-adversary pairs, with 1 indicating *agreement*, 0.5 *ambiguity*, and 0 *inconsistency*.

4.2 Results

Automatic Evaluation As shown in Table 2, F-ATTACK consistently achieves the highest attack success rate to attack LLaMA and has little impact on Mod and Sim, which indicates the effectiveness of F-ATTACK. Furthermore, F-ATTACK mostly obtains the best performance of modification and similarity metrics, except for AG's News, where F-ATTACK achieves the second-best. For instance, our framework only perturbs 4.1% of the words on the IMDB datasets, while the attack success rate is improved to 91.4% with a semantic similarity of 0.82. In general, our method can simultaneously satisfy the high attack success rate with a lower modification rate and higher similarity. Furthermore, We find that the attack success rate on document-level datasets, i.e., Yelp and IMDB, are higher than the other sentence-level datasets, which indicates that it is easier to mislead models when the input text is longer. The possible reason is the victim model tends to use surface clues rather than understand them to make predictions when the context is long. A case study is shown in Appendix G We also observe that F-ATTACK achieves a better attack effect on the binary classification task. Empirically, when there exist more than two categories, the impact of each replacement word may be biased towards a different class, leading to an increase in the perturbation rate.

Dataset		Con↑	Flu↑	Sim$_{\text{hum}}$ ↑
IMDB	Original	0.93	4.5	0.93
	F-ATTACK	0.87	4.1	
MR	Original	0.88	4.0	0.82
	F-ATTACK	0.79	3.8	

Table 3: Manual evaluation results comparing the original input and generated adversary by F-ATTACK of human prediction consistency (Con), language fluency (Flu), and semantic similarity (Sim$_{\text{hum}}$).

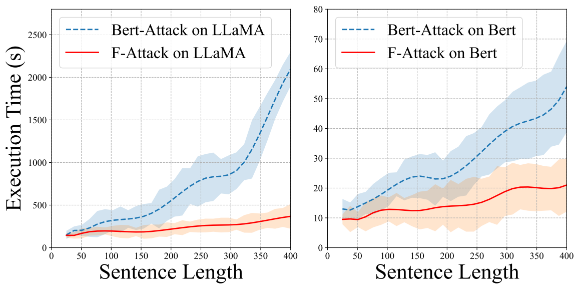

Figure 4: The time cost according to varying sentence lengths in the IMDB dataset. The left is on LLaMA while the right is on Bert.

Manual evaluation Following Fang et al., 2023, in manual evaluation, we first randomly select 100 samples from successful adversaries in IMDB and MR datasets and then ask ten crowd-workers to evaluate the quality of the original inputs and our generated adversaries. The results are shown in Table 3. As for the human prediction consistency, we regard the original inputs as a baseline. Taking IMDB as an example, humans can correctly judge 93% of the original inputs while they can maintain 87% accuracy to our generated adversaries, which indicates F-ATTACK can mislead the LLMs while keeping human judges unchanged. The language fluency scores of adversaries are close to the original inputs, where the gap scores are within 0.3 on both datasets. Furthermore, the semantic similarity scores between the original inputs and our generated adversaries are 0.93 and 0.82 in IMDB and MR, respectively. In general, F-ATTACK can satisfy the challenging demand of preserving the three aforementioned properties. Detailed design of manual evaluation and more results are shown in appendix F.

5 Analysis

5.1 Transferability

We evaluate the transferability of F-ATTACK samples to detect whether the samples generated from F-ATTACK can effectively attack other models. We conduct experiments on IMDB datasets and use the attacking accuracy of transferred sample from Bert-Attack as a baseline. As shown in Table 4, F-ATTACK has decrease transferred Accuracy across different models, which are consistently lower than Bert-Attack.

Compared to samples generated by traditional attack methods, the new samples generated by F-ATTACK can lower the accuracy by over 10 percentage on binary classification tasks, essentially confusing the Victim model greatly. Even a powerful baseline like ChatGPT would drop to only 68.6% accuracy. These samples incur very low computational costs and do not require attacks on specific Victim models.

	CNN	LSTM	Bert	LLaMA	Baichuan
CNN	+1.1	-16.9	-29.4	-18.7	-20.0
LSTM	-11.2	+1.9	-15.6	-22.9	-14.7
Bert	-10.7	-13.0	-1.1	-12.7	-12.8
LLaMA	-12.0	-13.4	-20.9	-8.2	-12.2
Baichuan	-11.1	-10.3	-11.2	-10.8	-2.2
ChatGPT	-7.4	-9.3	-11.2	-11.2	-9.4
Average	-10.6	-11.8	-17.6	-15.0	-13.8

Table 4: Transferability evaluation of F-ATTACK Samples on IMDB dataset. Each element is calculated from the difference in accuracy between F-ATTACK and Bert-Attack. Row i and column j is the Accuracy difference of samples generated from model j and evaluated on model i. The Average result is from non-diagonal elements of each column

5.2 Efficiency

In this section, we probe the efficiency according to varying sentence lengths in the IMDB dataset as shown in Figure 4. The time cost of F-ATTACK is surprisingly mostly better than Bert-Attack, which mainly targets obtaining cheaper computation costs with lower attack success rates in Table 2. Furthermore, with the increase of sentence lengths, F-ATTACK, and Bert-Attack maintain a stable time cost, while the time cost of BERT-attack is exploding. F-ATTACK has the advantage of much faster parallel substitution, hence as the sentence grows, the increase in time cost will be much smaller. These phenomena show the efficiency advantage of F-ATTACK, especially in dealing with long texts.

5.3 Generalization

Table 5 show that F-ATTACK not only has better attack effects against WordCNN and WordLSTM, but also misleads Bert and Baichuan, which are more robust models. For example, on the IMDB datasets, the attack success rate is up to 92.5% against Baichuan with a modification rate of only about 11.8% and a high semantic similarity of 0.75. Furthermore, the model generated by the Victim model created a decrease in accuracy to 71.4% on various black-box models of different scales.

Victim models	A-rate↑	Mod↓	Sim↑	Trans↓
WordCNN	96.3	9.1	0.84	78.5
WordLSTM	92.8	9.3	0.85	75.1
Bert	90.6	9.9	0.81	70.2
LLaMA*	91.8	13.1	0.74	68.6
Baichuan	92.5	11.8	0.75	71.4

Table 5: The results of F-Attack against other models.

Dataset	Acc↑	A-rate↑	Mod↓	Sim↑
Yelp	97.4	81.3	8.5	0.73
+Adv Train	95.9	65.7	12.3	0.67
IMDB	97.2	86.1	4.6	0.81
+Adv Train	95.5	70.2	7.3	0.78
SST-2	97.1	89.7	14.3	0.85
+Adv Train	92.2	68.6	16.8	0.83

Table 6: The results of comparing the original training with adversarial training with our generated adversaries. More results can be found in Appendix E.

5.4 Adversarial Training

We further investigate to improve the robustness of victim models via adversarial training. Specifically, we fine-tune the victim model with both original training datasets and our generated adversaries and evaluate it on the same test set. More details is on Appendix E.

As shown in Table 6, compared to the results with the original training datasets, adversarial training with our generated adversaries can maintain close accuracy, while improving performance on attack success rates, modification rates, and semantic similarity.The victim models with adversarial training are more difficult to attack, which indicates that our generated adversaries have the potential to serve as supplementary corpora to enhance the robustness of victim models.

5.5 Against Defense

Recently, **Entropy threshold defense** (Yao et al., 2023) has been used to defense against the attack on LLMs. It employs the entropy of the first token prediction to refuse responding. Figure 5 demonstrates the probability of top-10 tokens in the first generated word of LLaMA. It can be observed that the raw prompt usually generates the first token with low entropy (*i.e.*, the probabilty of argmax token is much higher, and the other tokens' probability is much lower).

As shown in Figure 5, the adversarial samples from F-Attack perform better than Bert-Attack with higher entropy. Attack samples generated through F-Attack fare better against entropy-based filters compared to traditional text adversarial attack methods, indicating that the samples created by F-Attack are harder to defend against.

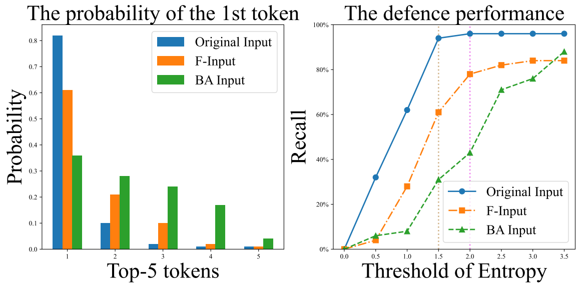

Figure 5: The subfigure (a) shows the probability of top-10 tokens in the first generated word in SA-LLaMA. The subfigure (b) demonstrates the defense performance with various entropy thresholds

6 Conclusion

In this paper, we examine the challenges encountered when traditional text adversarial attacks are applied to LLMs, such as lower success rates, slower attack speeds, and limited transferability of generated adversarial samples. Our investigation introduces the concept of *Important Level*, replacing the conventional *Important Score* module that contributed to slower attack speeds. Through the utilization of Prompts, we classify words based on their importance levels, significantly accelerating attack speeds. Furthermore, our approach involves employing various perturbation types, leading to enhanced transferability of generated adversarial samples. These modified samples exhibit improved applicability to other models, including opaque models like ChatGPT. Additionally, we incorporate a simple yet effective defense mechanism, utilizing entropy to assess whether input sentences qualify as adversarial samples. Our experiments validate that generated adversarial samples of F-Attack surpass this defense filter more adeptly, showcasing heightened linguistic fluency and deceptive capabilities, in line with our manual assessments. The experiments reaffirm that our strategy enhances speed, precision, and sample transferability when executing adversarial attacks on LLMs. Furthermore, while our methodology exhibits improvements in smaller models, it highlights the adaptability and effectiveness of our approach across diverse model scales.

Limitations

Our experiments were solely conducted on six selected datasets for two NLP tasks, all of which were English corpora. Furthermore, due to resource constraints in LLM research, our experimental results primarily relied on fine-tuning LLaMA-2-7B and Baichuan-2-7B as the base, without exploring LLMs or other open-source base models. Consequently, we lack evaluations on other types of LLMs, such as ELECTRA (Clark et al., 2020), XLNET (Yang et al., 2019) and other LLMs. Hence, our work lacks validation in terms of generalization and transferability across multi-task, multi-model, and multi-lingual aspects.

Ethics Statement

We declare that this article is in accordance with the ethical standards of *ACL Code of Ethics*. Any third-party tools used in this work are licensed by their authors. All crowd-workers participating in the experiments are paid according to the local hourly wages.

Acknowledgements

Thanks.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*.

David F Armstrong, William C Stokoe, and Sherman E Wilcox. 1995. *Gesture and the nature of language*. Cambridge University Press.

Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE security and privacy workshops (SPW)*, pages 1–7. IEEE.

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*.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. *arXiv preprint arXiv:1805.06130*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Claude Coulombe. 2018. Text data augmentation made simple by leveraging nlp cloud apis. *arXiv preprint arXiv:1812.04718*.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. *arXiv preprint arXiv:1806.09030*.

Xuanjie Fang, Sijie Cheng, Yang Liu, and Wei Wang. 2023. Modeling adversarial attack on pre-trained language models as sequential decision making. *arXiv preprint arXiv:2305.17440*.

Jules Gagnon-Marchand, Hamed Sadeghi, Md Haidar, Mehdi Rezagholizadeh, et al. 2019. Salsa-text: self attentive latent space based adversarial text generation. In *Canadian Conference on Artificial Intelligence*, pages 119–131. Springer.

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. IEEE.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*.

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

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. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.

Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems.

Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. 2018. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 195–231. Springer.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2020a. Contextualized perturbation for textual adversarial attack. *arXiv preprint arXiv:2009.07502*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Kai Liu, Xin Liu, An Yang, Jing Liu, Jinsong Su, Sujian Li, and Qiaoqiao She. 2020. A robust adversarial training approach to machine reading comprehension. In *AAAI*.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. 2023. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, page 100017.

John X Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*.

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*.

Jesús Oliva, José Ignacio Serrano, María Dolores Del Castillo, and Ángel Iglesias. 2011. Symss: A syntax-based measure for short-text semantic similarity. *Data & Knowledge Engineering*, 70(4):390–405.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

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.

Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. *arXiv preprint arXiv:1805.02917*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Michael Studdert-Kennedy. 2005. How did language go discrete. *Language origins: Perspectives on evolution, ed. M. Tallerman*, pages 48–67.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Xilie Xu, Keyi Kong, Ning Liu, Lizhen Cui, Di Wang, Jingfeng Zhang, and Mohan Kankanhalli. 2023. An llm can fool itself: A prompt-based adversarial attack. *arXiv preprint arXiv:2310.13345*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, and Li Yuan. 2023. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469*.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of nlp models. *arXiv preprint arXiv:2109.00544*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2019. Word-level textual adversarial attacking as combinatorial optimization. *arXiv preprint arXiv:1910.12196*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv e-prints*, pages arXiv–2306.

A Other Related Work

A.1 Sample Transferability

Evaluating text adversarial attacks relies heavily on sample transferability. It gauges how well attack samples perform across varied environments and models, measuring their broad applicability and consistency. In experiments, samples from the Victim-1 model directly target the Victim-2 model, testing transferability. Strongly transferable attack samples can hit almost all models in a black-box manner, a feat traditional white-box attacks can't match. Evaluation datasets like Adv-Glue's adversarial tasks showcase this transferability, aiding in robustness assessment.

In this paper, we use the Local Attack Success Rate (L-ASR) to show how well attack samples made by the Victim Model work on the Local Model. Remote Attack Success Rate (R-ASR) indicates their success on other models when directly transferred.

A.2 Synchronization Work

Prompt-Attack(Xu et al., 2023) leverages the exceptional comprehension of LLMs and diverges from traditional adversarial methods. It employs a manual approach of constructing rule-based prompt inputs, requiring LLMs to output adversarial attack samples that can deceive itself and meet the modification rule conditions. This attack method achieved fully automatic and efficient generation of attack samples using the local model. However, the drawback is that the model may not perform the whole attacking process properly, resulting in mediocre attack effectiveness. Additionally, different prompts can significantly influence the quality of the model-generated attack samples. While these generated attack samples possess some transferability, they do not delve into the internal reasoning of model, leading to the issue of excessively high modification rates.

Our work shares similarities in that we both leverage the understanding capabilities of LLMs. However, we only use language abstraction ability of ChatGPT to provide suggestions for Important Level, and the attack process still falls within the category of traditional text adversarial attack methods. Furthermore, our work is more focused on enhancing the transferability of attack samples and speeding up the attack, making them more widely applicable. This differs from the motivation of Prompt-Attack, which aims at the automated generation of samples that can deceive the model itself using LLMs.

B Attack Algorithm

The attacking process is shown in Algorithm 1. Since a_t^f is chosen through a probability distribution, the method is encouraged to explore more possible paths of substitutions. The instant result r_t is obtained from victim model after once substitution actions.

Once the attack finish condition is meet, the method will terminate this current step and output the answer to check whether it is succeeded. The expected return of substitutions is defined as follows:

$$J(\theta) = \mathbb{E}[G(\tau)] \tag{3}$$

Thus the result is calculated by ouput of model and can be expressed as follows:

$$\nabla J(\theta) = \frac{1}{M}\sum_{m=1}^{M} \nabla \log \pi_\theta(\tau^{(m)}) G(\tau^{(m)}) \tag{4}$$

where $[\tau^{(1)}, \tau^{(2)}, ..., \tau^{(M)}]$ are M samples of trajectories. The discount factor γ enables both long-term and immediate effects to be taken into account and trajectories with shorter lengths are encouraged.

We use all the test sets of each dataset and he average convergence time is approximately between 2-16 hours, related to the length of the input. When attacking large batches of samples, the impact of training cost is negligible compared to the cumulative attack time cost. During training, We adopt random strategies and short-sighted strategies in the initial stage for early exploration and to obtain better seeds.

C Datasets

We conduct experiments on the following datasets of Text Classification and detailed statistics are displayed in Table 7:

- **Yelp**(Zhang et al., 2015): A dataset for binary sentiment classification on reviews, constructed by considering stars 1 and 2 negative, and 3 and 4 positive.
- **IMDB**: A document-level movie review dataset for binary sentiment analysis.
- **MR**(Pang and Lee, 2005): A sentence-level binary classification dataset collected from Rotten Tomatoes movie reviews.

Algorithm 1 F-ATTACK Algorithm

```
Initialization: agent $\pi_\theta$ with parameters $\theta$, $\beta$, sentences number $M$
for $i \leftarrow 1$ to M do
    using ensemble-prompt on GPT-4 to get Important Level $L$
    while not receive termination signal do
        for $t \leftarrow 1$ to K do
            get words $s_t 1 \backsim s_t \beta$ from level-t
            compute $\pi_\theta((a_t^f, a_t^s)|s_t) \backsim \pi_\theta(a_t^f|s_t)$
            Using SimCSE to get Possible Substitutes $s_t 1 \backsim s_t \beta$
            compute reward $r_t$
            update $t \leftarrow t + 1$
        end for
    end while
    initialize $G(\tau) \leftarrow 0$
    for $j \leftarrow T$ to 1 do
        $G(\tau) \leftarrow \gamma G(\tau) + r_j$
        accumulate $J_j(\theta)$
    end for
    update $\theta \leftarrow \theta + \alpha \nabla J(\theta)$
end for
```

Dataset	Train	Test	Avg Len	Classes
Yelp	560k	38k	152	2
IMDB	25k	25k	215	2
AG's News	120k	7.6k	73	4
MR	9k	1k	20	2
SST-2	7k	1k	17	2

Table 7: Overall statistics of datasets.

- **AG's News**(Zhang et al., 2015): A collection of news articles. There are four topics in this dataset: World, Sports, Business, and Science/Technology.

- **SST-2** (Socher et al., 2013): The Stanford Sentiment Treebank task originates from reviews and is a binary sentiment classification dataset, where the task is to determine whether a given sentence conveys a positive or negative sentiment.

D Implementation Constraint

In order to make the comparison fairer, we set the following constraints for F-ATTACK as well as all baselines: (1) **Max modification rate**: To better maintain semantic consistency, we only keep adversarial samples with less than 40% of the words to be perturbed. (2) **Part-of-speech (POS)**: To generate grammatical and fluent sentences, we use NLTK tools[2] to filter candidates that have a different POS from the target word. This constraint is not employed on BERT-Attack. (3) **Stop words preservation**: the modification of stop words is disallowed and this constraint helps avoid grammatical errors. (4) **Word embedding distance**: For Textfooler, we only keep candidates with word embedding cosine similarity higher than 0.5 from synonyms dictionaries (Mrkšić et al., 2016). For *mask-fill* methods, following BERT-Attack, we filter out antonyms (Li et al., 2020b) via the same synonym dictionaries for sentiment classification tasks and textual entailment tasks.

Dataset	Acc↑	A-rate↑	Mod↓	Sim↑
Yelp	97.4	81.3	8.5	0.73
+Adv Train	95.9	65.7	12.3	0.67
IMDB	97.2	86.1	4.6	0.81
+Adv Train	95.5	70.2	7.3	0.78
AG-NEWS	95.3	77.1	15.3	0.83
+Adv Train	85.1	75.3	23.3	0.61
MR	95.9	83.2	11.1	0.53
+Adv Train	91.7	71.8	14.6	0.67
SST-2	97.1	89.7	14.3	0.85
+Adv Train	92.2	68.6	16.8	0.83

Table 8: Adversarial training results.

E Tuning with Adversaries

Table 8 displays adversarial training results of all datasets. Overall, after fine-turned with both original training datasets and adversaries, victim model is more difficult to attack. Compared to original results, accuracy of all datasets is barely affected, while attack success rate meets an obvious decline. Meanwhile, attacking model with adversarial training leads to higher modification rate, further demonstrating adversarial training may help improve robustness of victim models.

F Supplementary Results

At the beginning of manual evaluation, we provided some data to allow crowdsourcing workers to unify the evaluation standards. We also remove the data with large differences when calculating the average value to ensure the reliability and accuracy of the evaluation results. More manual evaluation results are shown in Table 9.

[2] https://www.nltk.org/

Dataset		**Con**↑	**Flu**↑	**Sim**$_{\text{hum}}$ ↑
IMDB	Original	0.96	4.6	1.00
	TextFooler	0.86	4.1	0.85
	Bert-Attack	0.82	4.3	0.91
	F-ATTACK	0.91	4.6	0.91
AG's News	Original	0.85	4.6	1.00
	TextFooler	0.77	3.9	0.84
	Bert-Attack	0.78	3.7	0.84
	F-ATTACK	0.76	4.3	0.81

Table 9: Manual evaluation results comparing the original input and generated adversary by attack method of human prediction consistency (Con), language fluency (Flu), and semantic similarity (Sim$_{\text{hum}}$).

G Case Study

Table 10 shows adversaries produced by F-ATTACK and the baselines. Overall, the performance of F-ATTACK is significantly better than other methods. For this sample from the MR dataset, only TextFooler and F-ATTACK successfully mislead the victim model, i.e., changing the prediction from *negative* to *positive*. However, TextFooler modifies twice as many words as the F-ATTACK, demonstrating our work has found a more suitable modification path. Adversaries generated by TextFooler and BERT-Attack are failed samples due to low semantic similarity. BERT-Attack even generates an invalid word "*enamoted*" due to its sub-word combination algorithm. We also ask crowd-workers to give a fluency evaluation.

Results show F-ATTACK obtains the highest score of 4 as the original sentence, while other adversaries are considered difficult to understand, indicating F-ATTACK can generate more natural sentences.

H Multi-Disturb

Following (Xu et al., 2023), we use 9 ways of disturbance, including character-level, word-level, and sentence-level disturbances as follows:

Method	**Text** (MR; Negative)	**Result**	**Mod**↓	**Sim**↑	**Flu**↑
Original	Davis is so enamored of her own creation that she can not see how insufferable the character is.	-	-	-	5
TextFooler	Davis is well enamored of her own infancy that she could not admire how infernal the idiosyncrasies is.	*Success*	33.3	0.23	3
BERT-Attack	Davis is often enamoted of her own generation that she can not see how insuffoure the queen is.	*Failure*	27.8	0.09	2
F-ATTACK	Davis is so charmed of her own crekation that she can~~'t~~ see how indefensible the character is. @kjdjq2.	*Success*	14.6	0.59	4

Table 10: Adversaries generated by F-ATTACK and baselines in MR dataset. The replaced words are highlighted in blue. *Failure* indicates the adversary fails to attack the victim model and *success* means the opposite.

Level	Abbre.	Perturbation Details
Character	C1	Choose at most two words in the sentence, and add letter to have typos.
	C2	Change at most two letters in the sentence.
	C3	Add at most two extraneous punctuation marks to the end of the sentence.
Word	W1	Replace at most two words in the sentence with synonyms.
	W2	Delete at most two words in the sentence with synonyms.
	W3	Add at most two semantically neutral words to the sentence.
Sentence	S1	Add a randomly generated short meaningless handle like @fasuv3.
	S2	Change the syntactic structure and word order of the sentence.
	S3	Paraphrase the sentence with ChatGPT.

Perturbation level	<sample>	Label → Prediction
Character (*C1*)	**Original**: less dizzying than just dizzy, the jaunt is practically over before it begins. **Adversarial**: less dizzying than just di~~z~~xzy, the jaunt is practically over before it begins.	negative → positive
Character (*C3*)	**Original**: if you believe any of this, i can make you a real deal on leftover enron stock that will double in value a week from friday. **Adversarial**: if you believe any of this, i can make you a real deal on leftover enron stock that will double in value a week from friday. :)	negative → positive
Word (*W2*)	**Original**: if you believe any of this, i can make you a real deal on leftover enron stock that will double in value a week from friday. **Adversarial**: if you believe any of this, i can make you a real deal on leftover enron stock that will double in value a week ~~from friday~~.	negative → positive
Word (*W3*)	**Original**: when leguizamo finally plugged an irritating character late in the movie. **Adversarial**: when leguizamo finally effectively plugged an irritating character late in the movie.	negative → positive
Sentence (*S2*)	**Original**: green might want to hang onto that ski mask, as robbery may be the only way to pay for his next project. **Adversarial**: green should consider keeping that ski mask, as it may provide the necessary means to finance his next project.	negative → positive
Sentence (*S3*)	**Original**: with virtually no interesting elements for an audience to focus on, chelsea walls is a triple-espresso endurance challenge. **Adversarial**: despite lacking any interesting elements for an audience to focus on, chelsea walls presents an exhilarating triple-espresso endurance challenge.	negative → positive