Evaluating the Validity of Word-level Adversarial Attacks with Large Language Models

Huichi Zhou[†] [1] **Zhaoyang Wang**[†] [2] **Hongtao Wang**[*] [1]
Dongping Chen[3] **Wenhan Mu**[4] **Fangyuan Zhang**[1]
Department of Computer, North China Electric Power University[1]
Sun Yat-sen University[2] HUST[3] Chongqing University[4]
{huichizhou77,luckychizuo,dongpingchen0612@}@gmail.com
whmu@stu.cqu.edu.cn {wanght,fyz}@ncepu.edu.cn

Abstract

Deep neural networks exhibit vulnerability to word-level adversarial attacks in natural language processing. Most of these attack methods adopt synonymous substitutions to perturb original samples for crafting adversarial examples while attempting to maintain semantic consistency with the originals. Some of them claim that they could achieve over 90% attack success rate, thereby raising serious safety concerns. However, our investigation reveals that many purportedly successful adversarial examples are actually invalid due to significant changes in semantic meanings compared to their originals. Even when equipped with semantic constraints such as BERTScore, existing attack methods can generate up to 87.9% invalid adversarial examples. Building on this insight, we first curate a 13K dataset for adversarial validity evaluation with the help of GPT-4. Then, an open-source large language model is fine-tuned to offer an interpretable validity score for assessing the semantic consistency between original and adversarial examples. Finally, this validity score can serve as a guide for existing adversarial attack methods to generate valid adversarial examples. Comprehensive experiments demonstrate the effectiveness of our method in evaluating and refining the quality of adversarial examples.

1 Introduction

Despite the success of Deep Neural Networks (DNNs) in various Natural Language Processing (NLP) tasks such as Text Classification, there is a line of research showing the vulnerability of DNNs to adversarial attacks (Alzantot et al., 2018; Ren et al., 2019; Li et al., 2020; Jin et al., 2020). These attacks involve introducing human imperceptible perturbations to original sentences, known as adversarial examples, in order to mislead victim models into making incorrect predictions. According to the definition of a successful adversarial attack, the generated adversarial example must satisfy two re-

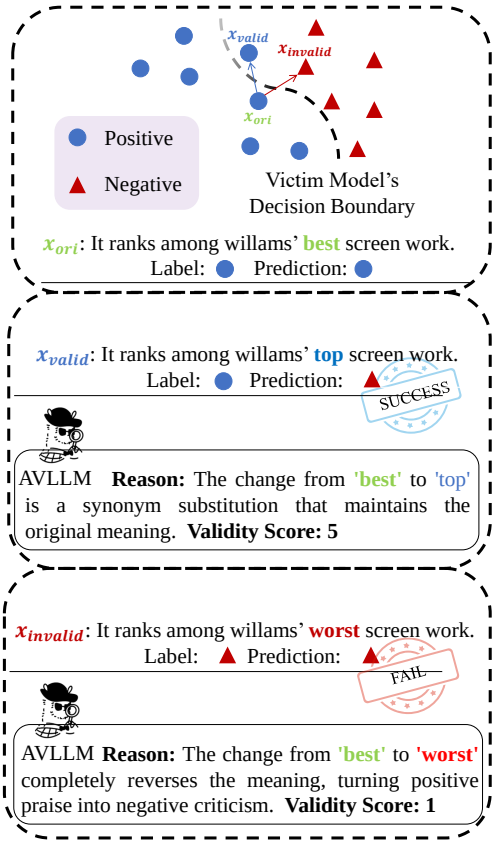


Figure 1: Demonstration of valid and invalid adversarial examples. A word-level attack method perturbs original sample x_{ori} to generate two different adversarial ones: x_{valid} and x_{invalid}. x_{valid} successfully retains its semantic meaning with x_{ori}, representing a **valid adversarial example**. However, x_{invalid} has a flip in semantic meaning, which fails to preserve semantic consistency to x_{ori}, representing an **invalid adversarial example**. Our *AVLLM* can offer scores and explanations of the validity for those generated adversarial examples by CoT reasoning. Then, validity scores can be used to help judge the success of adversarial attacks.

[†] Equal contribution.
[*] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 4902–4922
August 11-16, 2024

quirements. (1) **Misclassification**: victim model's prediction should be changed. (2) **Semantic Consistency**: the semantic meaning should remain sufficiently close to its original sentence.

The most effective adversarial attack against text classification DNNs is word-level attack, which substitutes a few words to generate perturbed adversarial examples, e.g., synonym substitutions (Garg and Ramakrishnan, 2020; Li et al., 2020, 2021; Ren et al., 2019; Alzantot et al., 2018; Yoo and Qi, 2021a). However, according to investigations in this paper, we find that these attack methods often fail to guarantee high semantic consistency and may even result in semantic flipping. As shown in Figure 1, two adversarial examples x_{valid} and x_{invalid} are generated over original sample x_{ori} with synonymous substitutions. Both of them have altered victim model's predictions from ● to ▲, meeting the first requirement, i.e., Misclassification. For the second requirement, i.e., Semantic Consistency, x_{valid} successfully retains its semantic meaning to x_{ori} (● → ●), while x_{invalid}'s meaning significantly changed (● → ▲)[1], leaving it an invalid adversarial example and a failed attack.

Numerous sentence embedding tools, semantic metrics and constraints, like grammar checkers, WordNet (Pedersen and Kolhatkar, 2009), USE (Cer et al., 2018), BERTScore (Zhang et al., 2019) and so on have been applied to adversarial attacks, in order to help word-level attack methods satisfy the semantic consistency requirement. However, as illustrated in Table 1, even equipped with constraints and having good performance in semantic metrics (e.g., in terms of BERTScore, all attack methods are $\geq 94\%$), these attack methods can still generate massive invalid adversarial examples (e.g., the real ASR (ASR^h) of BAE (Garg and Ramakrishnan, 2020) is only 11.7% instead of so-called 99.6%) according to the evaluations of human. This observation not only reveals limitations of current adversarial evaluations, but also indicates that existing constraints are not sufficient to help generate valid adversarial examples.

At the same time, we find that the validity results offered by GPT-4 (Achiam et al., 2023) (ASR^g) are highly consistent with human evaluations, which suggests the impressive ability of large language models (LLMs) in semantic understanding and evaluation. Based on this finding, we propose a

[1]That means the ground truth label of x_{invalid} is actually ▲, and the victim model made a correct prediction of it.

Attack Method	ASR	B.S.	Sim.	USE	ASR^h	ASR^g
A2T (Yoo and Qi, 2021b)	80.1	95.7	95.6	86.1	$45.2_{\downarrow 34.9}$	$39.7_{\downarrow 40.4}$
BAE (Garg and Ramakrishnan, 2020)	99.6	95.1	94.4	86.9	$11.7_{\downarrow 87.9}$	$13.3_{\downarrow 88.3}$
PWWS (Ren et al., 2019)	87.6	94.7	95.5	86.3	$29.0_{\downarrow 58.6}$	$29.1_{\downarrow 58.5}$
Textfooler (Jin et al., 2020)	98.8	96.0	95.2	86.5	$17.8_{\downarrow 81.0}$	$20.0_{\downarrow 78.8}$
F-alzantot (Jia et al., 2019)	94.3	95.3	95.4	85.8	$28.8_{\downarrow 65.5}$	$35.3_{\downarrow 59.0}$

Table 1: Pilot study for evaluating the validity of adversarial examples. ASR denotes attack success rate. Metrics like BERTScore (B.S.), SimCSE (Sim.), and USE assess similarity between original and adversarial examples. ASR^h and ASR^g are validity rates of adversarial examples judged by human and GPT-4. The details of these metrics are introduced in Sec. 4.1.

novel method named as **A**dversarial **V**alidity evaluation with **L**arge **L**anguage **M**odels (*AVLLM*) to address the issue of invalid adversarial examples. Specifically, we first collect a 13K word-level adversarial validity evaluation dataset with rich annotations by GPT-4. Then, with recent advancements of open-source LLMs, we fine-tune a lightweight LLM to provide a validity score for assessing the semantic consistency between original and adversarial examples. Also, Chain-of-Thought (Wei et al., 2022) (CoT) is leveraged to enable our *AVLLM* offer detailed explanations of semantic differences, making the validity score interpretable. In addition, the lightness of *AVLLM* makes it possible to be integrated into most existing adversarial attack methods, serving as a plug-and-play module to help generate valid adversarial examples. Extensive experiments across various datasets demonstrate *AVLLM*'s superior performance in evaluating and refining the quality of adversarial examples.

In conclusion, our contributions are three-fold:

1) Our investigations reveal that many popular adversarial attack methods have serious issues in generating invalid adversarial examples, and existing semantic constraints or sentence embedding tools are not sufficient to help with it.

2) A 13K dataset is curated by leveraging the impressive semantic understanding ability of LLMs for word-level adversarial validity evaluation, which can facilitate the research of textual adversarial attack.

3) We propose *AVLLM*, as a interpretable metric for adversarial validity evaluations, and as a plug-and-play module to help generate valid adversarial examples. Extensive experiments show its effectiveness in evaluating and refining the quality of adversarial examples.

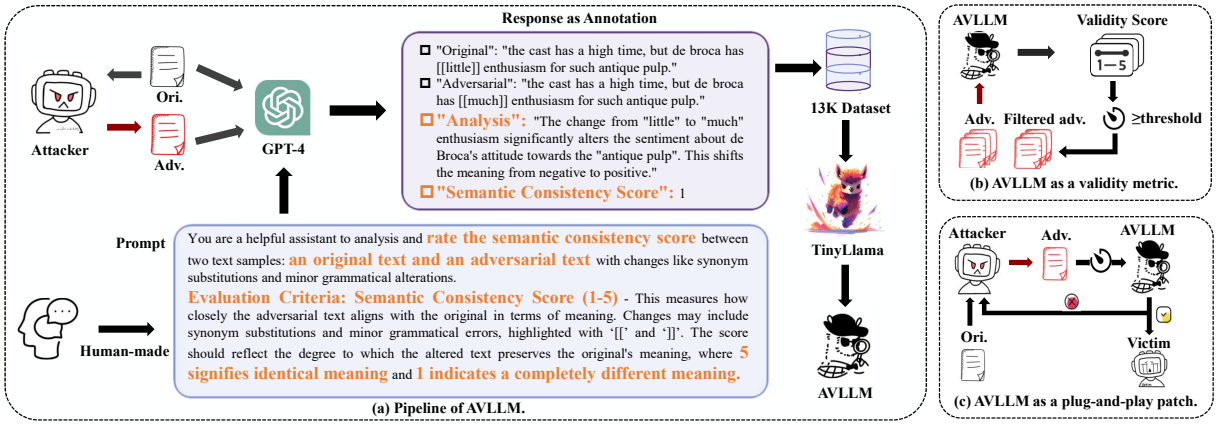


Figure 2: The framework of AVLLM. (a) We collect a dataset that comprises 13K word-level adversarial examples across a various of attack methods and datasets. Then, GPT-4 is instructed as a semantic evaluator to provide scores and analysis for the dataset. Finally, TinyLlama (Zhang et al., 2024) is selected as a lightweight LLM to fine-tune on this dataset, resulting in our *AVLLM*. (b) To assess the validity of generated adversarial examples and revise the existing benchmark, *AVLLM* is serving as a adversarial validity metric. (c) To guide the search and generation of valid adversarial examples, *AVLLM* is integrated into the adversarial attack method.

2 Related Works

Adversarial Attack According to the perturbation grains, adversarial attacks can be categorized into three levels: character-level, word-level, and sentence-level. Character-level attacks often utilize spelling errors to mislead victim models (Gao et al., 2018; Eger et al., 2019; Li et al., 2019). Sentence-level attacks often use generative adversarial networks or text paraphrase technologies to directly generate adversarial examples (Iyyer et al., 2018; Maheshwary et al., 2021; Lei et al., 2022). However, both character and sentence level attacks can easily cause grammar errors and largely modify the text structure, which make them challenging to maintain the quality of adversarial examples. Most recent research attempts to develop word-level attacks which often use gradient information (Guo et al., 2021; Yuan et al., 2021) or substitution strategies (Alzantot et al., 2018; Garg and Ramakrishnan, 2020; Li et al., 2020; Ren et al., 2019). However, our investigation reveals that word-level methods can generate massive invalid adversarial examples due to semantic consistency issue, which has not been well discussed in previous works.

Adversarial Defense To defend against adversarial attacks, various defense methods have been proposed including data augmentation (Ng et al., 2020; Si et al., 2021; Kober et al., 2021), adversarial training (Wang and Bansal, 2018; Shafahi et al., 2019; Zhu et al., 2020; **?**), and reconstruction-based methods (Jones et al., 2020; Xu et al., 2022; Wang et al., 2023). There is also a line of research that focuses on detection of adversarial examples (Mosca et al., 2022; Huber et al., 2022; Raina and Gales, 2022) Notably, the proposed *AVLLM* aims at evaluating the validity of adversarial examples and refining their quality, instead of defending against adversarial attacks.

Evaluation of Adversarial Examples Existing word-level adversarial attack methods evaluate the quality of their generated adversarial examples by synonyms and semantic constraints, such as (1) substitution limitations by NLTK (Bird and Loper, 2004), WordNet (Pedersen and Kolhatkar, 2009) and counter-fitting (Mrkšić et al., 2016), (2) sentence similarity scores by modification rate, ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2019), (3) embedding distance by USE (Cer et al., 2018) and SimCSE (Gao et al., 2021). However, our investigation reveals that these automatic evaluation metrics are not sufficient to reflect the semantic consistency between original and adversarial examples. Previous studies also conduct human evaluations to assess the quality of adversarial examples (Alzantot et al., 2018; Jin et al., 2020) focusing more on the grammar, fluency and readability instead of semantic consistency which is our focus. A recent line of research attempts to evaluate the semantic consistency by human judges (Morris et al., 2020a; Herel et al., 2022; Dyrmishi et al., 2023) which may introduce human bias and can be considerably costly. In contrast to evaluation by combinations of embedding distance and seman-

tic metrics (Morris et al., 2020a; Chiang and Lee, 2023), we propose to leverage impressive semantic understanding ability of LLMs to provide a interpretable metric for the adversarial validity.

3 Method

3.1 Preliminary

Given a text classification dataset, each sample is a pair of input text $x \in \mathcal{X}$ and its corresponding label $y \in \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output space, respectively. We consider a well-trained victim model f that maps $\mathcal{X}$ to $\mathcal{Y}$. For an original sample x correctly predicted by the victim model, i.e., $f(x) = y$, an adversarial attack method may perturb x to generate an adversarial example x_{adv}, which should satisfy the following conditions:

$$f(x_{\text{adv}}) \neq f(x) \text{ and } \mathcal{C}(x_{\text{adv}}, x) \geq \epsilon, \quad (1)$$

where $\mathcal{C}$ is a semantic consistency function (e.g., cosine similarity), and ϵ denotes the threshold differentiating between the original sample and the adversarial example. There are two conditions in Eq.(1): (1) Misclassification: $f(x_{\text{adv}}) \neq f(x)$, and (2) Semantic Consistency: $\mathcal{C}(x_{\text{adv}}, x) > \epsilon$. Adversarial examples typically satisfy the first condition, but they often fall short of meeting the second condition, which we refer to as "Validity" in this paper.

As listed in Table 2, a generated adversarial example should first maintain its semantic meaning with the original sample, thereby having the same semantic label, i.e., $y_{\text{adv}} = y$. Subsequently, the victim model should make a false prediction on the adversarial example, i.e., $f(x_{\text{adv}}) \neq f(x)$. Once these criteria are met, this adversarial example x_{adv} can be deemed as a successful adversarial example.

3.2 *AVLLM*

Towards judging whether the adversarial example satisfies $y_{\text{adv}} = y$, prior works often use various semantic metrics mostly relying on embedding distance and similarity (e.g., USE), which are not sufficient as the semantic consistency function $\mathcal{C}$. In this paper, we propose to leverage LLMs to provide a score with an explanation for assessing the validity of adversarial examples. The obtained score can then serve as the semantic consistency function. Figure 2 shows that our method can be divided into 3 parts: pipeline of *AVLLM*, serving as a validity metric, and serving as a plug-and-play patch.

Note that for integrating into existing word-level attack process, it is necessary to train an open-source, lightweight and specific *AVLLM*, other than using a closed-source, compute-heavy, and general LLM (such as GPT-4). Compared with calling GPT-4 API, adopting *AVLLM* can save costs, and boost the inference speed due to its fewer parameters.

3.3 Pipeline of *AVLLM*

Dataset Construction To facilitate research on evaluating the validity of adversarial examples, we collect a dataset consisting of about 13K samples with semantic consistency explanations annotated by GPT-4. The dataset statistics are shown in Table 3. Specifically, we first sample a total of 15K data from 5 popular text classification datasets, including AGNEWS[2], IMDB (Maas et al., 2011a), MR (Zhang et al., 2015a), SST-2 (Socher et al., 2013), and YELP[3]. Considering that the attack and annotation costs are related to the length of sentence, we sampled from these datasets inversely proportional to the length. Then, these samples are uniformly distributed to 3 adversarial attack methods, including TextFooler (Jin et al., 2020), BERT-Attack (Li et al., 2020), and PWWS (Ren et al., 2019) for attacking. We choose BERT (Devlin et al., 2019)[4] as the victim model. Through pilot experiments, we observe that GPT-4[5] is able to provide the semantic consistency score that is highly consistent with human evaluations. Specifically, the score consistency rate between GPT-4 and human with a tolerance of ± 1, is 0.99. The Mann–Whitney U test (McKnight and Najab, 2010) also confirms the differences in these scores are not statistically significant with $p < 0.05$. Thus, we instruct GPT-4 to annotate each adversarial example with a score ranging from 1 to 5, and an explanation as illustrated in Figure 2(a). After filtering out invalid ones, we finally obtain a 13.7K adversarial validity evaluation dataset split into training and validation sets with 12K and 1.7K samples, respectively. More details are in Appendix A.

Fine-tune Recent advancements in open-source LLMs (Touvron et al., 2023; Li et al., 2023; Biderman et al., 2023; Zhang et al., 2024) make it possible to leverage a lightweight LLM for adver-

[2] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

[3] https://www.yelp.com/dataset

[4] https://huggingface.co/google-bert/bert-base-uncased

[5] GPT-4-1106-preview is used in this paper.

y	y_{adv}	$f(x_{\text{adv}})$	Mis.	Validity	Success
●	●	●	✗	✓	✗
●	●	▲	✓	✓	✓
●	▲	●	✓	✗	✗
●	▲	▲	✗	✗	✗

Table 2: The relationship in a binary sentiment classification task among the label of original sample ('y'), the label of adversarial example ('y_{adv}'), victim model's mis-prediction of the adversarial one ('Mis.'), a valid adversarial example ('Validity'), and a successful attack ('Success'). ● and ▲ represent the positive sentiment and negative sentiment, respectively.

Dataset	Avg. Len	# Pre-sampling	# Success
AGNEWS	37	2000 (13.1%)	1806 (13.1%)
IMDB	234	300 (2.0%)	198 (1.4%)
MR	21	6000 (39.2%)	5691 (41.2%)
SST2	10	6000 (39.2%)	5253 (38.1%)
YELP	133	1000 (6.5%)	849 (6.2%)
Total	-	15300 (100.0%)	13797 (100.0%)

Table 3: Statistics of the curated Dataset. "Success" refers to the successful generated adversarial examples. Numbers in parentheses are the percentage of samples.

sarial validity evaluation, which should be more efficient and cost-effective than using GPT-4. In this paper, we choose TinyLlama (Zhang et al., 2024), a compact and open-source 1.1B LLM as our *AVLLM*. However, smaller LLMs often struggle with complex tasks requiring CoT reasoning, which indicates that TinyLlama cannot provide corresponding explanation e for the validity score VS.

To enhance the interpretability of the score, we propose to fine-tune it on the adversarial validity evaluation dataset for specialized use. We optimize the supervised fine-tuning objective as follows:

$$\mathcal{L} = \mathbb{E} \log \mathrm{P}_{\text{AVLLM}} \left([e, \text{VS}] \,|\, [\texttt{Inst}, x, x_{\text{adv}}] \right), \quad (2)$$

where `Inst` is the template shown in Table 11.

After fine-tuning, *AVLLM* is expected to generate an accurate validity score VS to assess the semantic consistency with a detailed explanation. The explanation can not only boost the score accuracy thanks to CoT reasoning, but also help understand the semantic differences between the original and adversarial examples. The score is serving as the semantic consistency function $\mathcal{C}$:

$$\mathcal{C}(x_{\text{adv}}, x) = \text{VS}(x_{\text{adv}}, x). \quad (3)$$

3.4 *AVLLM* as a Validity Metric

Through our pilot experiments, we find that many purportedly successful adversarial examples are not valid due to the semantic consistency issue. To evaluate the real performance of attack methods, we propose to use validity score as an evaluation metric. This evaluation is conducted after the adversarial examples are generated. The revised attack success rate (ASR^{ϵ}) is calculated as follows:

$$\text{ASR}^{\epsilon} = \frac{|S \wedge \{x_{\text{adv}} \in S : \mathcal{C}(x_{\text{adv}}, x) \geq \epsilon\}|}{|\{(x, y) \in \mathcal{X} \times \mathcal{Y} : f(x) = y\}|}, \quad (4)$$

where S is the set of successful adversarial examples before filtering. ϵ is the semantic difference threshold which can be empirically set according to the validity score distribution. The adversarial examples that do not meet this threshold are considered as invalid and failed.

Algorithm 1 *AVLLM* as a patch.

1: **Input:** Text-label pair (x, y), victim model f, semantic consistency function $\mathcal{C}$, semantic difference threshold ϵ
2: **Output:** A valid adversarial example x_{adv}
3: **Initialization:** $x_{\text{adv}} = x$
4: $\mathbf{W} \leftarrow \varnothing$
5: **for** $1 \leq i \leq |x|$ **do**
6: $w \leftarrow$ important word ranking
7: $\mathbf{W} \leftarrow \mathbf{W} \cup \{w\}$
8: **end for**
9: **for** each word w_j in $\mathbf{W}$ **do**
10: Initiate the set of CANDIDATES by extracting synonyms using different methods
11: **for** each c_k in CANDIDATES **do**
12: $x' \leftarrow$ Replace w_j with c_k in x_{adv}
13: **if** $C(x_{\text{adv}}, x) \geq \epsilon$ **then**
14: **if** $f(x_{\text{adv}}) \neq y$ **then return** x_{adv}
15: **else**
16: Adjusting the ranking of W
17: **end if**
18: **end if**
19: **end for**
20: **end for**
21: **return** NONE

3.5 *AVLLM* as a Plug-and-play Patch

To guide existing attack methods effectively searching valid adversarial examples, we propose to integrate *AVLLM* into the adversarial attack process, serving as a plug-and-play patch. As outlined in Algorithm 1, most word-level adversarial attack methods can be summarized into two main steps: (1) Word Importance Ranking (L. 3-8). The attack method adopts various ranking strategies to build its word substitution candidate set. (2) Generating Adversarial Example (L. 9-17). The attack method attempts to craft an adversarial example by replacing the word with the highest importance score with its candidate. Unlike existing approaches, the crafted adversarial example is not directly tested by the victim model. Instead, we take a simple

Dataset/ Methods		LSTM						BERT						ALBERT					
		ASR			Semantic			ASR			Semantic			ASR			Semantic		
		ASR^o	ASR^2	ASR^3	B.S.	USE	VS	ASR^o	ASR^2	ASR^3	B.S.	USE	VS	ASR^o	ASR^2	ASR^3	B.S.	USE	VS
AGNEWS	A2T	59.1	$31.5_{\downarrow 27.6}$	$12.2_{\downarrow 46.9}$	96.5	98.0	2.02	56.2	$24.6_{\downarrow 31.6}$	$7.5_{\downarrow 48.7}$	96.5	98.2	1.83	49.1	$47.3_{\downarrow 1.8}$	$20.3_{\downarrow 28.8}$	96.4	98.3	1.99
	BAE	99.7	$19.6_{\downarrow 80.1}$	$2.7_{\downarrow 97.0}$	95.8	97.8	1.37	96.8	$15.5_{\downarrow 81.3}$	$1.3_{\downarrow 95.5}$	94.8	97.9	1.30	94.6	$18.0_{\downarrow 76.6}$	$2.6_{\downarrow 92.0}$	95.1	98.1	1.34
	PWWS	80.1	$24.0_{\downarrow 56.1}$	$7.1_{\downarrow 73.0}$	95.3	96.2	1.62	66.0	$21.7_{\downarrow 44.3}$	$7.2_{\downarrow 58.8}$	95.2	96.7	1.65	72.2	$39.6_{\downarrow 32.6}$	$12.3_{\downarrow 59.9}$	95.6	97.5	1.74
	Alzantot	95.3	$36.5_{\downarrow 58.8}$	$13.4_{\downarrow 81.9}$	95.1	96.4	1.77	90.4	$37.2_{\downarrow 53.2}$	$11.2_{\downarrow 79.2}$	94.9	95.9	1.76	90.8	$38.3_{\downarrow 52.5}$	$13.5_{\downarrow 77.3}$	95.4	96.9	1.83
	TextFooler	98.5	$31.2_{\downarrow 67.3}$	$6.5_{\downarrow 92.0}$	95.8	97.0	1.56	93.3	$23.3_{\downarrow 70.0}$	$5.9_{\downarrow 87.4}$	94.9	96.9	1.47	92.7	$26.4_{\downarrow 66.3}$	$7.0_{\downarrow 85.7}$	95.4	97.6	1.52
IMDB	A2T	92.9	$62.5_{\downarrow 30.4}$	$41.1_{\downarrow 51.8}$	96.7	98.2	2.57	75.3	$49.3_{\downarrow 26.0}$	$19.2_{\downarrow 56.1}$	95.7	97.5	2.24	71.8	$53.4_{\downarrow 18.4}$	$19.0_{\downarrow 52.8}$	95.8	95.6	2.36
	BAE	100.0	$21.4_{\downarrow 78.6}$	$10.7_{\downarrow 89.3}$	97.2	98.5	1.45	100.0	$16.4_{\downarrow 83.6}$	$1.4_{\downarrow 98.6}$	96.3	98.3	1.28	98.8	$35.1_{\downarrow 63.7}$	$5.3_{\downarrow 93.5}$	96.2	98.6	1.57
	PWWS	96.4	$53.6_{\downarrow 42.8}$	$23.2_{\downarrow 73.2}$	96.4	97.3	2.20	94.5	$45.2_{\downarrow 49.3}$	$20.6_{\downarrow 73.9}$	96.1	98.0	1.94	78.4	$48.5_{\downarrow 29.9}$	$26.3_{\downarrow 52.1}$	95.9	97.6	2.36
	Alzantot	96.4	$53.6_{\downarrow 42.8}$	$32.1_{\downarrow 64.3}$	95.9	96.6	2.24	97.3	$58.9_{\downarrow 38.4}$	$26.0_{\downarrow 71.3}$	95.2	96.7	2.14	86.0	$64.3_{\downarrow 21.7}$	$31.6_{\downarrow 54.4}$	95.3	96.7	2.56
	TextFooler	100.0	$60.7_{\downarrow 39.3}$	$21.4_{\downarrow 78.6}$	96.9	98.3	2.20	100.0	$57.5_{\downarrow 42.5}$	$19.2_{\downarrow 80.8}$	95.6	97.2	2.00	95.3	$57.9_{\downarrow 37.4}$	$24.0_{\downarrow 71.3}$	95.7	97.0	2.14
MR	A2T	93.3	$71.8_{\downarrow 21.5}$	$33.7_{\downarrow 59.6}$	96.2	98.5	2.48	71.8	$54.0_{\downarrow 17.8}$	$19.6_{\downarrow 52.2}$	95.8	97.9	2.35	71.8	$50.3_{\downarrow 21.5}$	$17.8_{\downarrow 54.0}$	95.8	97.9	2.33
	BAE	99.4	$41.7_{\downarrow 57.7}$	$12.9_{\downarrow 86.5}$	96.5	99.2	1.72	96.3	$27.0_{\downarrow 69.3}$	$4.9_{\downarrow 91.4}$	95.8	97.9	1.44	98.8	$37.4_{\downarrow 61.4}$	$5.9_{\downarrow 92.9}$	96.2	99.0	1.56
	PWWS	90.2	$58.3_{\downarrow 31.9}$	$29.5_{\downarrow 60.7}$	96.1	98.0	2.36	79.8	$44.8_{\downarrow 35.0}$	$18.4_{\downarrow 61.4}$	95.7	97.6	2.17	78.4	$51.5_{\downarrow 26.9}$	$25.7_{\downarrow 52.7}$	95.9	97.7	2.38
	Alzantot	93.9	$71.8_{\downarrow 22.1}$	$36.2_{\downarrow 57.7}$	96.1	98.0	2.67	87.1	$57.1_{\downarrow 30.0}$	$20.9_{\downarrow 66.2}$	95.1	97.8	2.22	86.0	$62.6_{\downarrow 23.4}$	$31.0_{\downarrow 55.0}$	95.3	97.6	2.53
	TextFooler	98.8	$66.9_{\downarrow 31.9}$	$33.1_{\downarrow 65.7}$	96.3	98.7	2.39	93.9	$50.3_{\downarrow 43.6}$	$16.0_{\downarrow 77.9}$	95.2	96.9	1.95	95.3	$57.3_{\downarrow 38.0}$	$21.6_{\downarrow 73.7}$	95.7	97.9	2.11
SST2	A2T	94.9	$69.5_{\downarrow 25.4}$	$26.6_{\downarrow 68.3}$	96.0	98.1	2.36	71.5	$46.2_{\downarrow 25.3}$	$19.9_{\downarrow 51.6}$	95.4	96.8	2.27	72.4	$49.0_{\downarrow 23.4}$	$15.1_{\downarrow 57.3}$	94.7	96.3	2.16
	BAE	100.0	$36.2_{\downarrow 63.8}$	$4.5_{\downarrow 95.5}$	96.5	98.5	1.54	98.4	$26.3_{\downarrow 72.1}$	$2.2_{\downarrow 96.2}$	96.1	98.0	1.37	99.5	$22.4_{\downarrow 77.1}$	$2.6_{\downarrow 96.9}$	96.1	98.6	1.39
	PWWS	89.8	$55.9_{\downarrow 33.9}$	$23.2_{\downarrow 66.6}$	96.2	97.7	2.20	81.7	$43.0_{\downarrow 38.7}$	$16.7_{\downarrow 65.0}$	95.2	95.9	2.02	81.8	$46.4_{\downarrow 35.4}$	$22.9_{\downarrow 58.9}$	95.8	97.5	2.20
	Alzantot	93.2	$72.9_{\downarrow 20.3}$	$36.7_{\downarrow 56.5}$	95.7	97.5	2.57	80.7	$53.2_{\downarrow 27.5}$	$19.4_{\downarrow 61.3}$	95.0	96.4	2.21	80.7	$53.2_{\downarrow 27.5}$	$21.0_{\downarrow 59.7}$	95.0	96.1	2.25
	TextFooler	100.0	$66.7_{\downarrow 33.3}$	$28.3_{\downarrow 71.7}$	96.2	98.1	2.25	99.5	$51.6_{\downarrow 47.9}$	$17.2_{\downarrow 82.3}$	95.1	96.3	1.97	97.9	$57.3_{\downarrow 40.6}$	$20.8_{\downarrow 77.1}$	95.6	97.2	2.07
YELP	A2T	89.6	$65.3_{\downarrow 24.3}$	$32.4_{\downarrow 57.2}$	96.0	96.7	2.44	72.3	$39.3_{\downarrow 33.0}$	$15.2_{\downarrow 57.1}$	95.5	96.1	2.05	53.7	$34.6_{\downarrow 19.1}$	$15.4_{\downarrow 38.3}$	95.2	93.5	2.32
	BAE	98.8	$16.4_{\downarrow 82.4}$	$2.9_{\downarrow 95.9}$	96.9	98.4	1.32	99.5	$14.7_{\downarrow 84.8}$	$0.5_{\downarrow 99.0}$	95.7	98.0	1.25	97.9	$13.3_{\downarrow 84.6}$	$2.1_{\downarrow 95.8}$	96.2	98.2	1.25
	PWWS	91.9	$46.2_{\downarrow 45.7}$	$13.9_{\downarrow 78.0}$	96.2	95.7	1.87	89.5	$37.7_{\downarrow 51.8}$	$13.6_{\downarrow 75.9}$	95.8	94.9	1.81	88.3	$28.7_{\downarrow 59.6}$	$10.6_{\downarrow 77.7}$	95.6	94.5	1.68
	Alzantot	93.1	$62.4_{\downarrow 30.7}$	$28.3_{\downarrow 64.8}$	95.7	96.7	2.36	95.8	$55.0_{\downarrow 40.8}$	$18.3_{\downarrow 77.5}$	95.2	94.9	2.08	95.8	$58.1_{\downarrow 37.7}$	$20.4_{\downarrow 75.4}$	95.2	94.9	2.11
	TextFooler	97.7	$60.1_{\downarrow 37.6}$	$17.3_{\downarrow 80.4}$	96.2	97.0	2.07	97.9	$49.2_{\downarrow 48.7}$	$16.8_{\downarrow 81.1}$	95.2	96.0	1.97	97.9	$54.8_{\downarrow 43.1}$	$19.2_{\downarrow 78.7}$	95.2	95.6	2.06

Table 4: Re-benchmark of 5 attack methods against victim models including LSTM (Hochreiter and Schmidhuber, 1997), BERT (Devlin et al., 2019), and ALBERT (Lan et al., 2019) on 5 text classification datasets. Each attack method dynamically generate adversarial examples to target each victim model with 1,000 test samples.

yet effective modification to the attack process (L. 13). The adversarial example is first evaluated by our semantic consistency function with the help of *AVLLM*, and only if it meets the semantic difference threshold, next steps will be proceeded. The impact of this modification is highlighted. The rejection of invalid adversarial example is expected to adjust the search direction[6], thus guiding attack method to generate valid and high-quality ones.

4 Experiment

4.1 Evaluation Metrics

There are several metrics used in this paper: (1) Original Attack Successful Rate (ASR^o) is the number of adversarial examples against the number of correctly predicted samples, without any post-revision. (2) Revised ASRs: ASR^h, ASR^g and ASR^ϵ represent original ASRs after filtering out invalid adversarial examples by human, GPT-4, and *AVLLM* with the threshold ϵ, respectively. (3) USE (Cer et al., 2018), BERTScore (B.S.) (Zhang et al., 2019), SimCSE (Sim.) (Gao et al., 2021) and language modeling Perplexity (PPL) calculated by GPT-2 (Radford et al., 2019) represent semantic metrics which are often used to measure the sentence similarity between original and adversarial examples. (4) Validity Score (VS) generated by *AVLLM* represents the average validity score of adversarial examples. Please refer to Appendix B for more experimental settings.

4.2 Validity Evaluation Results

The goal of validity evaluation is to assess the validity of generated adversarial examples and re-benchmark these attack methods. From results in Table 4, we have two main observations. (1) Same with prior works, ASR^o is very high. But there is a significant decrease (ASR^ϵ) when filtering out invalid examples by *AVLLM*, even with a tolerant threshold $\epsilon = 2$. This indicates that real attack success rates of these attack methods can be much lower than expected. (2) Existing metrics, such as B.S. and USE, are very high, ranging from 93.5 to 99. However, our VS scores remain below 3 out of 5 in average, as highlighted in gray background, indicating the low validity of generated adversarial examples. This discrepancy suggests that current sentence similarity metrics are not sufficient to effectively measure the semantic consistency between original samples and adversarial examples.

4.3 Online Attack Results via Patch

To test the performance of *AVLLM* as a patch being integrated into existing attack methods, we conduct experiments evaluating the attack performance after patching, with ASR and various se-

[6]The detailed adjustment is up to specific attack method.

	Metric	ASR(↑%)			USE(↑%)			BERTScore(↑%)			SimCSE(↑%)			∇ PPL(↓)			∇ Grammar(↓)		
	Threshold	ASR^o	ASR^3	ASR^4	ASR^o	ASR^3	ASR^4	ASR^o	ASR^3	ASR^4	ASR^o	ASR^3	ASR^4	ASR^o	ASR^3	ASR^4	ASR^o	ASR^3	ASR^4
AGNEWS	A2T	51.6	38.9	7.2	98.1	98.5	**99.9**	96.6	96.7	**96.8**	45.6	45.7	**50.2**	36.7	38.9	**6.9**	0.31	0.21	**0.09**
	BAE	98.9	40.0	29.5	96.7	99.7	**99.8**	94.3	96.3	**97.3**	45.8	**46.0**	45.9	49.0	17.6	**9.1**	1.03	0.55	**0.14**
	PWWS	74.7	36.8	21.1	96.3	97.9	**99.3**	94.6	96.4	**97.5**	44.8	45.0	**45.2**	46.3	65.9	**40.5**	0.83	0.51	**0.38**
	TextFooler	100.0	43.2	36.8	96.3	99.6	**99.7**	94.4	96.5	**96.7**	44.5	45.3	**45.6**	42.1	**22.9**	24.1	0.69	0.51	**0.26**
IMDB	A2T	75.3	53.4	39.7	97.5	97.8	**98.4**	96.6	96.7	**97.1**	57.8	57.7	**58.1**	13.2	**11.7**	21.3	0.29	**0.27**	**0.27**
	BAE	100.0	47.9	57.5	98.4	99.3	**99.7**	96.2	96.9	**97.6**	57.9	59.1	**59.8**	18.5	7.4	**4.0**	0.58	0.29	**0.26**
	PWWS	94.5	47.9	42.5	97.9	**99.2**	99.0	96.3	97.1	**97.3**	58.3	59.4	**60.8**	25.7	18.4	**6.9**	0.52	0.40	**0.38**
	TextFooler	100.0	52.1	50.7	97.3	99.3	**99.4**	95.6	96.9	**97.2**	57.8	59.0	**59.1**	36.9	24.8	**8.6**	0.59	0.24	**0.21**
MR	A2T	70.0	52.0	34.0	97.5	97.8	**99.7**	95.7	96.2	**96.9**	54.6	55.6	**55.6**	42.3	32.3	**12.1**	0.26	0.19	**0.00**
	BAE	95.0	54.0	22.0	97.9	99.2	**99.9**	95.7	96.5	**97.4**	53.5	53.8	**54.9**	32.1	28.1	**0.4**	0.09	**0.06**	**0.06**
	PWWS	78.0	45.0	18.0	96.9	98.5	**99.6**	95.1	96.0	**96.5**	54.5	55.7	**57.5**	28.6	31.6	**9.0**	**0.17**	0.20	**0.17**
	TextFooler	90.0	58.0	24.0	96.9	99.3	**99.9**	94.9	99.3	**99.9**	54.5	**55.2**	55.1	10.9	**0.2**	10.6	0.28	0.34	**0.17**
SST2	A2T	66.7	50.5	15.1	95.1	96.0	**99.1**	95.1	**96.1**	95.9	51.9	52.1	**56.0**	42.1	34.8	**12.6**	0.18	**0.11**	0.21
	BAE	98.9	82.8	60.2	93.5	95.8	**97.6**	96.0	96.2	**96.3**	50.6	51.5	**51.9**	41.2	**17.1**	29.9	0.09	0.10	**0.03**
	PWWS	77.4	45.2	38.7	94.3	96.7	**96.9**	94.9	95.8	**96.4**	52.5	**53.0**	52.1	43.6	44.6	**37.8**	**0.17**	0.19	**0.17**
	TextFooler	100.0	68.8	51.6	93.6	96.8	**97.9**	94.5	95.3	**96.1**	51.9	**52.6**	52.2	**52.2**	57.4	59.9	0.26	0.17	**0.16**
YELP	A2T	71.7	54.0	41.0	95.8	96.1	**97.0**	95.7	96.1	**97.0**	51.4	**52.2**	52.0	32.2	21.5	**16.6**	0.23	0.22	**0.17**
	BAE	99.0	68.0	42.0	96.0	97.5	**99.2**	95.9	95.6	**96.8**	50.7	51.8	**52.6**	19.7	20.2	**11.3**	0.62	0.58	**0.50**
	PWWS	85.6	47.0	38.0	94.5	96.3	**97.9**	95.5	96.1	**96.6**	52.1	52.1	**52.3**	23.6	**22.6**	57.5	0.59	0.53	**0.50**
	TextFooler	96.0	71.0	54.0	94.8	97.8	**98.6**	95.5	96.2	**96.7**	51.2	52.0	**52.2**	16.0	24.1	**6.5**	0.29	0.29	**0.22**

Table 5: Evaluating *AVLLM* as a patch (ASR^3 and ASR^4) across 4 attack methods against BERT model on 5 datasets. ∇PPL is the PPL difference between origin samples and adversarial examples. The best performance is in **bold**.

mantic consistency metrics. From Table 5, we find that: (1) Though the ASR after patching shows a decline, it overpasses offline results in Table 4, which indicates our *AVLLM* can effectively guide the search. (2) Our method consistently boosts nearly all semantic metrics. Meanwhile, as the constraint strength increases (↑ ϵ), the semantic consistency between adversarial examples and original samples becomes better. This shows the effectiveness of our method in helping attack methods generate valid adversarial examples. (3) In terms of ∇PPL, adversarial examples after patching are more grammatically coherent and semantically continuous, suggesting *AVLLM* as a patch is able to improve the quality of adversarial examples. (4) We also use grammar checker[7] to detect grammar errors additionally introduced by adversarial attacks (∇Grammar). The results suggest our method can limit these caused grammar errors in generating adversarial examples, helping maintain the semantic consistency and fluency of the text.

5 Analysis

In this section, we aim to take comprehensive analysis and engage in thorough discussions to study the effectiveness of our method in evaluating and refining the quality of adversarial examples.

5.1 Credibility of Validity Score

We take consistency evaluations in Table 6 to verify the credibility of the proposed validity score. We find that exact match rates may not be very convincing, however, note that ϵ representing for the threshold of a valid or an acceptable adversarial example can be empirically tuned[8]. With a tolerance range of 1, the consistency rates for both GPT-4 and *AVLLM* compared to human is 0.99, suggesting the high credibility of the proposed validity score. Surprisingly, *AVLLM* exhibits a higher alignment with human, which indicates the success of our specialized fine-tuning.

	GPT-4	Human
GPT-4	-	0.72 / 0.99 / ±0.28
AVLLM	0.66 / 0.97 / ±0.39	0.86 / 0.99 / ±0.15

Table 6: Consistency rate of validity score between *AVLLM*, GPT-4, and human. The first figure in each cell is the exact match rate, the second figure is the consistency rate with a tolerance of 1, and the third figure is the mean absolute difference. TinyLlama w/o fine-tuning is hard to follow the instruction even under few-shot settings, thus being excluded from comparison.

5.2 Comparison with Existing Constraints

Table 7 compares the performance of our Validity Score constraint (VS) with existing semantic constraints. The results show that our validity constraint is more effective in helping attack methods generate adversarial examples with high semantic consistency. Besides, we find that for all semantic constraints, the performance gap ($ASR^o - ASR^g$) is much lower as the constraints become tighter. In

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

[8] That is the reason why this paper often shows experimental results with different ϵ values.

	Constraint	ASR^o	ASR^g	B.S.	USE	Time (sec)
MR	Default	**86.2**	$25.6_{\downarrow 60.6}$	95.6	95.3	**2.9**
	USE ≥ 0.90	25.6	$12.9_{\downarrow 12.7}$	97.4	96.4	3.4
	USE ≥ 0.95	4.7	$3.5_{\downarrow 1.2}$	98.2	97.3	7.3
	B.S. ≥ 0.90	42.4	$14.7_{\downarrow 27.6}$	97.4	92.8	6.9
	B.S. ≥ 0.95	22.9	$9.1_{\downarrow 13.8}$	**98.3**	94.3	8.1
	VS ≥ 2	74.1	$27.1_{\downarrow 47.0}$	88.0	96.0	26.7
	VS ≥ 3	52.3	$\mathbf{37.8}_{\downarrow 14.5}$	97.0	98.7	34.1
	VS ≥ 4	24.5	$19.8_{\downarrow 4.8}$	97.7	**99.8**	64.9
SST2	Default	**85.8**	$19.2_{\downarrow 66.6}$	95.4	95.7	**3.5**
	USE ≥ 0.90	25.8	$12.1_{\downarrow 13.7}$	97.5	96.3	4.3
	USE ≥ 0.95	9.2	$6.4_{\downarrow 2.8}$	**98.3**	**98.4**	6.8
	B.S. ≥ 0.90	44.6	$16.4_{\downarrow 28.2}$	97.5	91.6	5.7
	B.S. ≥ 0.95	23.7	$9.9_{\downarrow 13.8}$	**98.3**	93.6	7.6
	VS ≥ 2	76.3	$26.1_{\downarrow 50.2}$	86.1	96.1	34.9
	VS ≥ 3	61.8	$\mathbf{42.5}_{\downarrow 19.3}$	95.6	96.3	36.3
	VS ≥ 4	41.4	$33.2_{\downarrow 8.2}$	96.2	97.9	57.6

Table 7: Comparison of the performance among various semantic constraints. "Default" means the default constraint setting of attack methods. "Time" means the average time of generating a successful adversarial example. The performance is averaged over 4 attack methods including A2T, BAE, PWWS and TextFooler.

terms of real attack performance (ASR^g), our validity score constraint achieves the best performance when the threshold is set to 3. Even we adopt a lightweight LLM as *AVLLM*, it still brings much computational overhead. We will explore more efficient LLM technologies, such as quantization and parallelism, to improve its speed in the future.

5.3 Suggestions for Adversarial Attacks

In this section, we aim to provide valuable suggestions for both the evaluation of adversarial attacks and designing of attack methods.

Evaluation Evaluation is known to be time-consuming since word-level attack methods would search the whole space of substitution combinations to find a successful adversarial example, resulting in thousands queries for long sentence. According to Figure 3, most valid adversarial examples can be found within 152 queries if we consider the minimum semantic threshold is 3. Similarly, the modification rate of most valid adversarial examples is 0.31. This finding suggests that the evaluation can save lots of time by limiting the number of queries and the modification rate with appropriate values. Please refer to Appendix C for results of other datasets and attack methods.

Design of Attack Method The used constraints and sources of synonyms are crucial in designing an effective and semantic consistent adversarial attack method. As shown in Table 8, we find that (1) Masked language modeling (MLM) based attack method (e.g., BAE) demonstrates limitations in searching valid adversarial examples that satisfy the semantic consistency requirement, since the objective of MLM is predicting the masked word with most probable words rather than synonyms. For instance, when the attack method are attempting to find synonyms by predicting "`[MASK]`" in the sentence "`I [MASK] this movie`", both "like" and "hate," which are antonyms, are possible predictions. (2) Using WordNet (Pedersen and Kolhatkar, 2009) (e.g., PWWS) or counter-fitting (Mrkšić et al., 2016) (e.g., TextFooler) as the source of finding synonym candidates generally have higher validity scores, thus being recommended. (3) More constraints often lead to better performance. Typically, adversarial examples generated by A2T who uses 4 unique constraints has

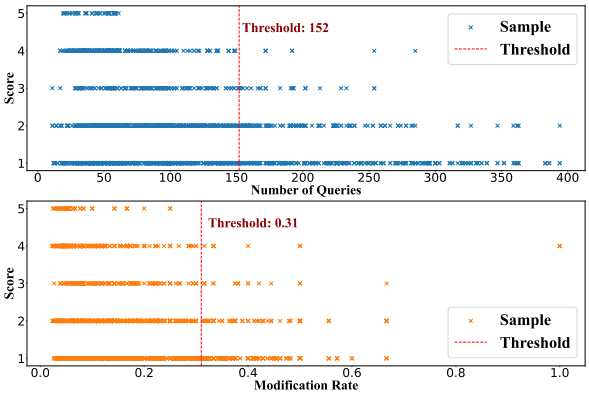


Figure 3: Correlation between validity score, number of queries, and modification rate on the MR dataset with TextFooler as the attack method. Thresholds are plotted based on 95% percentile of valid adversarial examples.

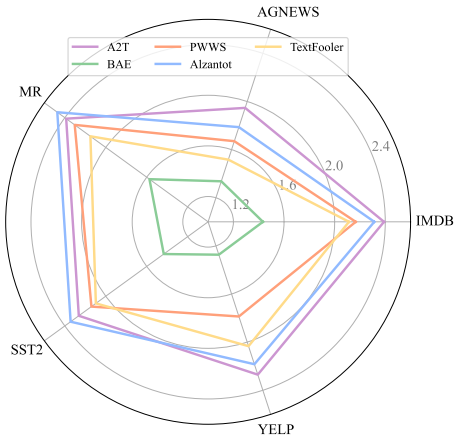


Figure 4: Performance in semantic consistency of 5 attack methods on 5 datasets.

Dataset	Attacker	Example
SST2	PWWS	another in-your-face wallow in the lower depths made by people who have never sung those blues. another in-your-face wallow in the **humbled** depths made by people who have never sung those blues. another in-your-face wallow in the **small** depths made by people who have never sung those blues.
IMDB	Textfooler	This is the greatest movie ever. If you have written it off with out ever seeing it. You must give it a second try. This is the **worst** movie ever. If you have written it off with out ever seeing it. You must give it a second try. This is the **noblest** movie ever. If you have written it off with out ever seeing it. You must give it a second try.
YELP	BAE	Nice atmosphere. Cheeseburger was not all that. Nice **too**. Cheeseburger was not all that. **Great** atmosphere. Cheeseburger was not all that.
YELP	BAE	Props to all the performers but it really wasn't worth seeing. Trust me, I've seen a lot of shows... Props to all the performers but it **sometimes** wasn't worth seeing. Trust me, I've seen a lot of **films**... Props to all the performers but it **almost** wasn't **needed** seeing. Trust me, I've seen a lot of **films**...

Table 9: Comparison of original and adversarial examples on different datasets under the PWWS and BAE attacks. In each cell of "Example", the first row represents the original sample, the second row is the adversarial example w/o *AVLLM*, and the third row is the adversarial example w/ *AVLLM*.

Attack Method	Constraints	Source of Synonyms
A2T	Modification Rate Word Embedding Distance DistilBERT Cosine Similarity Part-of-Speech Consistency	Counter-fitting Embedding
BAE	USE	Masked Language Modeling
Alzantot	Modification Rate PPL Word embedding distance	Counter-fitting Embedding
PWWS	-	WordNet
Textfooler	Word Embedding Distance Part-of-Speech Consistency USE	Counter-fitting Embedding

Table 8: Basic information of 5 attack methods including default constraints and sources of finding synonyms.

the highest validity score. This indicates developing more constraints is necessary for designing more powerful attack methods.

5.4 Case Study

The case study on the quality of the generated adversarial examples is shown in Table 9. Compared with directly generated adversarial examples (w/o *AVLLM*), with the help of *AVLLM*, the generated adversarial examples tend to have higher semantic consistency with the original samples, which indicates that the quality of adversarial examples can be improved with the proposed patch.

6 Conclusion

In this paper, our investigation first reveals a prevalent issue in word-level adversarial attack methods, wherein many of purportedly successful adversarial examples are actually invalid due to the semantic consistency issue. Building on this finding, we advocate using LLMs to offer an interpretable validity score for assessing the semantic consistency between original sample and adversarial example. To this end, we construct a 13K dataset for adversarial validity evaluation with the help of GPT-4, and then fine-tune a lightweight LLM as *AVLLM* for saving costs and boosting the inference speed. The proposed *AVLLM* can not only serve as a validity metric to assess the semantic consistency of adversarial examples, but also serve as a plug-and-play patch to help existing attack methods generate high-quality and valid adversarial examples. Through extensive experiments and analysis, our method demonstrates its efficacy and provides valuable insights for advancing research in adversarial attacks.

Limitations

- This paper focuses on word-level adversarial attacks. While these attacks are the most common in this field, the semantic consistency issues of other types of attacks such as sentence-level attack are valuable to study. Besides, the studied types of word-level attack algorithms are basically limited to synonyms substitutions, other types of substitutions like insertion and deletion are not well included.

- We only consider TinyLlama (Zhang et al., 2024) as *AVLLM*. It is necessary to explore more open-source LLMs to validate the effectiveness of our method. Also for the victim models, currently popular large-scale models like LLaMA (Touvron et al., 2023) are not being tested, while they are important to study especially in the era of LLM.

- According to our experiments in Table 7, our method may bring much computational overhead to the attack process. Recent LLM technologies for efficiency, such as quantization and parallelism will be explored in the future.

Ethics Statement

In our experiment, we utilized models and dataset that are entirely open-source except GPT-4, ensuring transparency and accessibility. Furthermore, our *AVLLM* incorporates enhanced adversarial attack capabilities, which enables the craft of valid adversarial examples that are more challenge to detect. The adversarial examples with high semantic consistency may bring additional risks to existing DNNs based applications.

Acknowledgments

We thank the anonymous reviewers for their insightful and valuable comments. We also extend our gratitude to Yiqiang Li, and the ENLP Laboratory for their assistance and support.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

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*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Are synonym substitution attacks really synonym substitution attacks? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1853–1878, Toronto, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Salijona Dyrmishi, Salah Ghamizi, and Maxime Cordy. 2023. How do humans perceive adversarial text? a reality check on the validity and naturalness of word-based adversarial attacks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8822–8836, Toronto, Canada. Association for Computational Linguistics.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5747–5757, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

David Herel, Hugo Cisneros, and Tomas Mikolov. 2022. Preserving semantics in textual adversarial attacks. *arXiv preprint arXiv:2211.04205*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computing*, 9(8):1735–1780.

Lukas Huber, Marc Alexander Kühn, Edoardo Mosca, and Georg Groh. 2022. Detecting word-level adversarial text attacks via SHapley additive exPlanations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 156–166, Dublin, Ireland. Association for Computational Linguistics.

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

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142, Hong Kong, China. Association for Computational Linguistics.

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

Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust encodings: A framework for combating adversarial typos. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2752–2765, Online. Association for Computational Linguistics.

Thomas Kober, Julie Weeds, Lorenzo Bertolini, and David Weir. 2021. Data augmentation for hypernymy detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1034–1048, Online. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yibin Lei, Yu Cao, Dianqi Li, Tianyi Zhou, Meng Fang, and Mykola Pechenizkiy. 2022. Phrase-level textual adversarial attack with label preservation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1095–1112, Seattle, United States. Association for Computational Linguistics.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069, Online. Association for Computational Linguistics.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS*. The Internet Society.

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

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011a. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011b. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021. A strong baseline for query efficient attacks in a black box setting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8396–8409, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick E McKnight and Julius Najab. 2010. Mann-whitney u test. *The Corsini encyclopedia of psychology*, pages 1–1.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020a. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*,

pages 3829–3839, Online. Association for Computational Linguistics.

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

Edoardo Mosca, Shreyash Agarwal, Javier Rando Ramírez, and Georg Groh. 2022. "that is a suspicious reaction!": Interpreting logits variation to detect NLP adversarial attacks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7806–7816, Dublin, Ireland. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1268–1283, Online. Association for Computational Linguistics.

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

Ted Pedersen and Varada Kolhatkar. 2009. WordNet::SenseRelate::AllWords - a broad coverage word sense tagger that maximizes semantic relatedness. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Demonstration Session*, pages 17–20, Boulder, Colorado. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Vyas Raina and Mark Gales. 2022. Residue-based natural language adversarial attack detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3836–3848, Seattle, United States. Association for Computational Linguistics.

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

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

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32.

Chenglei Si, Zhengyan Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2021. Better robustness by more coverage: Adversarial and mixup data augmentation for robust finetuning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1569–1576, Online. Association for Computational Linguistics.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yicheng Wang and Mohit Bansal. 2018. Robust machine comprehension models via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 575–581, New Orleans, Louisiana. Association for Computational Linguistics.

Zhaoyang Wang, Zhiyue Liu, Xiaopeng Zheng, Qinliang Su, and Jiahai Wang. 2023. RMLM: A flexible defense framework for proactively mitigating word-level adversarial attacks. In *Proceedings of the*

61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2757–2774, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jianhan Xu, Cenyuan Zhang, Xiaoqing Zheng, Linyang Li, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2022. Towards adversarially robust text classifiers by learning to reweight clean examples. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1694–1707, Dublin, Ireland. Association for Computational Linguistics.

Jin Yong Yoo and Yanjun Qi. 2021a. Towards improving adversarial training of NLP models. *CoRR*, abs/2109.00544.

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

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270.

Lifan Yuan, Yichi Zhang, Yangyi Chen, and Wei Wei. 2021. Bridge the gap between CV and nlp! A gradient-based textual adversarial attack framework. *CoRR*, abs/2110.15317.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015a. Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Dataset

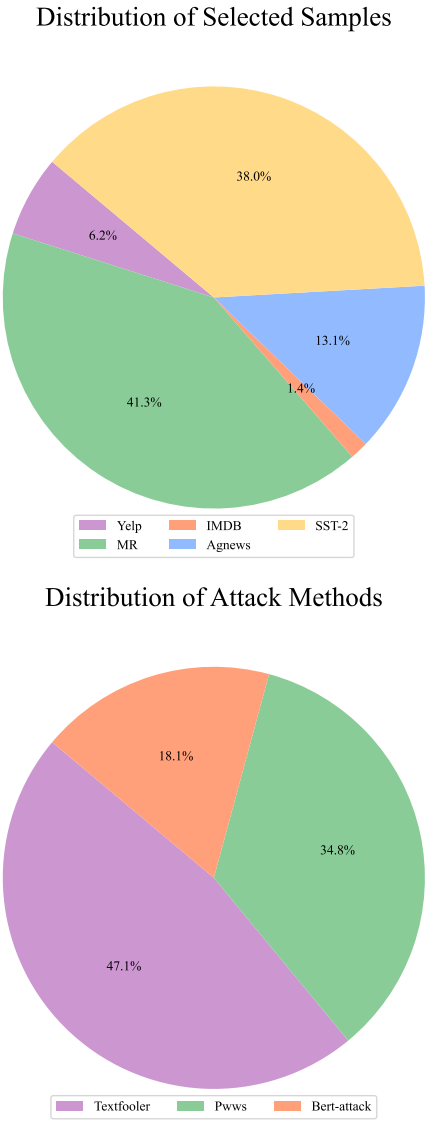


Figure 5: Upper - Dataset distribution of selected samples. Lower - Sample distribution by attack method.

In this paper, we utilize adversarial attack framework proposed by Morris et al. (2020b), which were generated using multiple adversarial attack module (Morris et al., 2020b) including adversarial examples crafted using techniques such as PWWS (Ren et al., 2019), TextFooler (Jin et al., 2020), BAE (Garg and Ramakrishnan, 2020) and so on. These examples were tested on a range of models, including LSTM (Yu et al., 2019), BERT (Devlin et al., 2019), and ALBERT (Lan et al., 2019), which were trained or fine-tuned on datasets like IMDB (Maas et al., 2011a), AGNEWS (Zhang et al., 2015b) and so on.

The victim models used to generate these

classifiers were fine-tuned using the TextAttack toolkit (Morris et al., 2020b) and publicly available on the TextAttack documentation page[9] and on the Huggingface model hub[10]. Detailed hyperparameters for fine-tuning these models are available in the respective model cards and `config.json` files.

Each dataset employed serves a specific purpose in our study: AGNEWS, an essay-level dataset, is used for multi-class news classification. IMDB, a document-level dataset, focuses on sentiment classifications of movie reviews. The MR dataset offers fine-grained labels for movie review sentiment classification. SST-2 enables phrase-level sentiment analysis, and the YELP dataset comprises business reviews used for sentiment analysis.

B Detailed Experiment Settings

B.1 Datasets

Experiments are conducted on five benchmark classification datasets from phase-level to document-level tasks, including AGNEWS (Zhang et al., 2015b), IMDB (Maas et al., 2011b), MR (Pang and Lee, 2005), SST2 (Socher et al., 2013) and YELP (Zhang et al., 2015a).

B.2 Victim Models

Three different types of DNNs are adopted as victim models, including long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997), $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019), and $\text{ALBERT}_{\text{BASE-V2}}$ (Lan et al., 2019). LSTM consists of 2 layers of 300-dimensional memory cells with the 300-dimensional pre-trained GloVe embeddings (Pennington et al., 2014). $\text{BERT}_{\text{BASE}}$ and $\text{ALBERT}_{\text{BASE-V2}}$ contains 12 layers of 768-dimensional transformer blocks and one linear layer for classification, but $\text{ALBERT}_{\text{BASE-V2}}$ has less parameter than $\text{BERT}_{\text{BASE}}$.

B.3 Attack Methods

Five strong word-level adversarial attack methods are employed as attackers. Yoo and Qi (2021b) propose A2T which first use gradient-based word importance ordering and DistilBERT (Sanh et al., 2019) semantic textual similarity constraint. Garg and Ramakrishnan (2020) first compute token importance and then replaces and inserts tokens in the original text by masking a portion of the text and leveraging the BERT-MLM to generate substitutions. Ren et al. (2019) propose PWWS which considers the word saliency to determine the word modification order for greedy attack. Jia et al. (2019) develop an attack algorithm that exploits population-based gradient-free optimization via genetic algorithms. Jin et al. (2020) first identify the important words and then replace them with the semantically similar and grammatically correct words, named TextFooler.

B.4 Inference Settings

Given that high temperature brings a better alignment with humans, for the stage of using GPT-4 for inference and collecting samples, we use a temperature of 0.95 and a top-p of 1; For the hyperparameters for TinyLlama, the temperature we use is 0.9 and the top-p is 0.95.

B.5 Finetune Settings

For the fine-tuning protocol, we adhered to standard hyperparameters known to yield effective results in model training. We set a training batch size of 16 per device, leveraging gradient accumulation over one step to enhance memory efficiency. A cosine decay strategy was selected for the learning rate scheduler, balancing the need for learning rate reduction over time while maintaining model adaptability. To monitor progress and ensure data integrity, logging and model saving were scheduled at intervals of every 5 and 500 steps, respectively. The initial learning rate was carefully chosen at $5e^{-5}$, with the training extending over 6 epochs to thoroughly imbibe the nuances of our 13K dataset into TinyLlama. Additionally, we utilized half-precision floating-point (fp16) training to quicken the training phase without compromising the model's learning capacity significantly. DeepSpeed optimization was also employed to further enhance training efficiency, reducing computational demand while maintaining high performance.

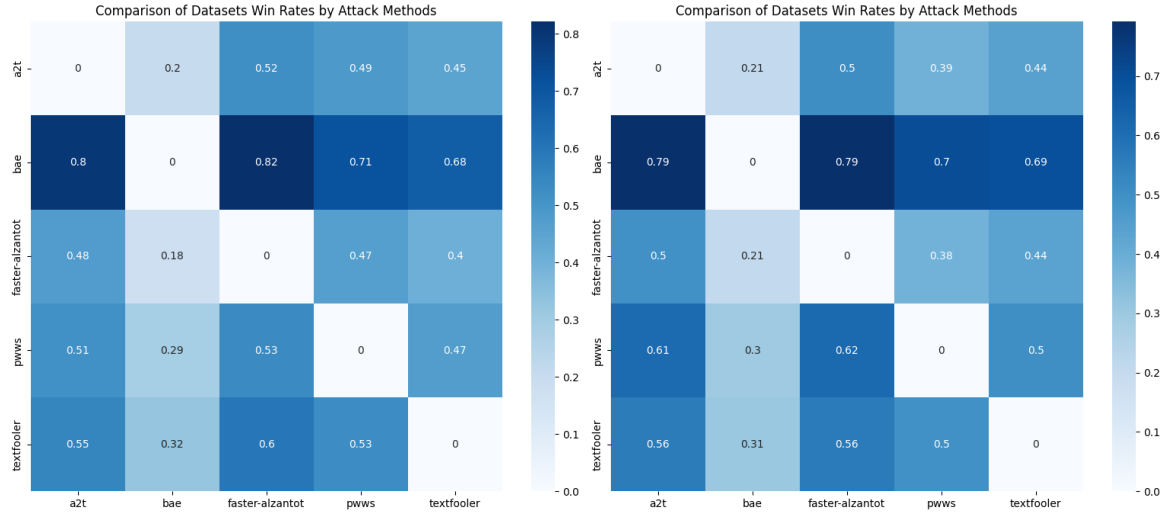
C Additional Experimental Results

C.1 Win-Rate

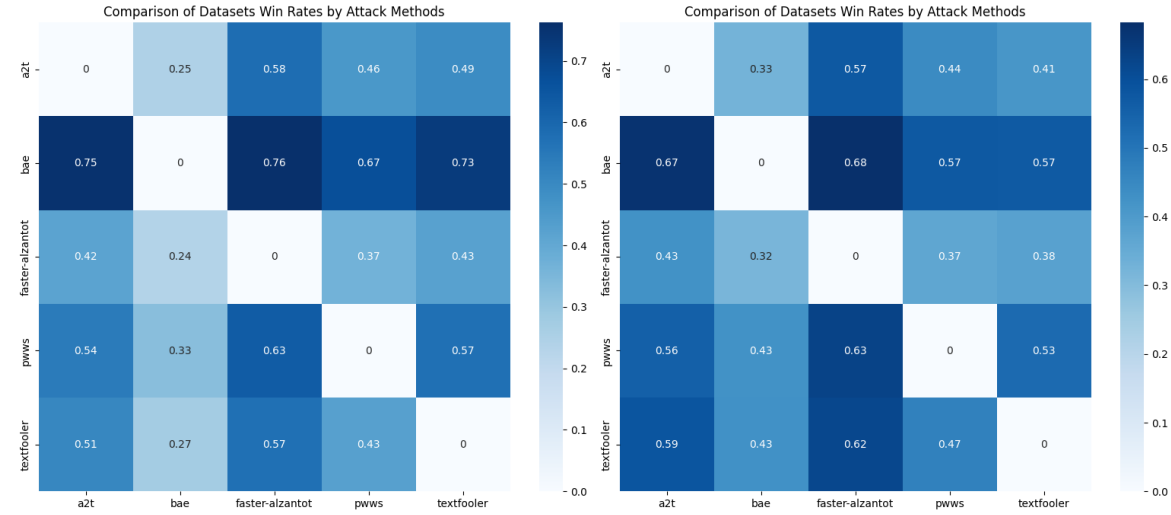
To evaluate which adversarial attack method have the best performance in remaining semantic similarity. We conduct a comprehensive experiment which evaluate the result of different adversarial attack methods. The first step, for each dataset, we randomly sample 100 test data from the orig-

[9] https://textattack.readthedocs.io/en/latest/3recipes/models.html

[10] https://huggingface.co/textattack



(a) MR (b) SST2



(c) YELP (d) AGNEWS

Figure 6: Win Rates by Adversarial Attack Methods in different datasets

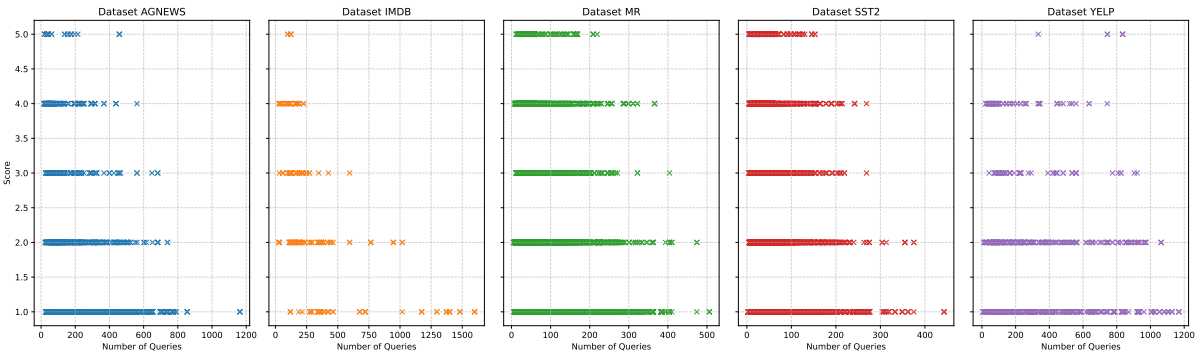


Figure 7: Scatter plots show score vs. query numbers across datasets. X-axis: query numbers; Y-axis: Score. Different subplots reveal varied query number ranges. 'AGNEWS' and 'IMDB' exhibit a strong negative correlation between increasing query numbers and Score. 'MR' and 'YELP' show a similar but weaker correlation. 'SST2' displays samples in a narrow query number range with no discernible correlation.

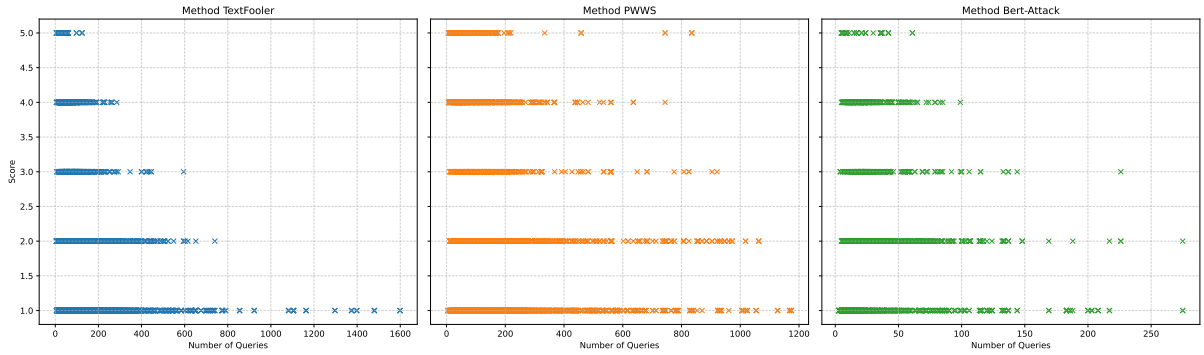
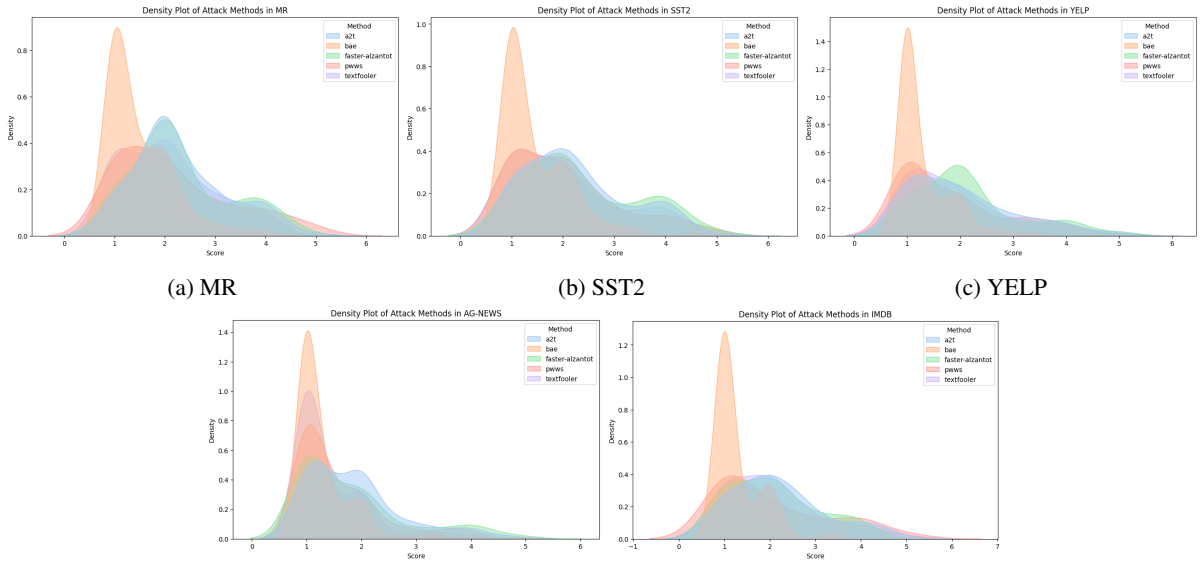


Figure 8: Scatter plots display Score vs. query numbers for different attack methods. X-axis: query numbers; Y-axis: Score. Subplots show varying query number ranges. 'Textfooler' and 'PWWS' methods exhibit a wider range of query numbers with a noticeable negative correlation with Score. 'Bert-Attack' method samples mainly fall within a smaller query number range ($\leq$100) with no evident correlation between query numbers and Score.



(a) MR (b) SST2 (c) YELP

(d) AGNEWS (e) IMDB

Figure 9: The density plot of adversarial attack methods in different dataset

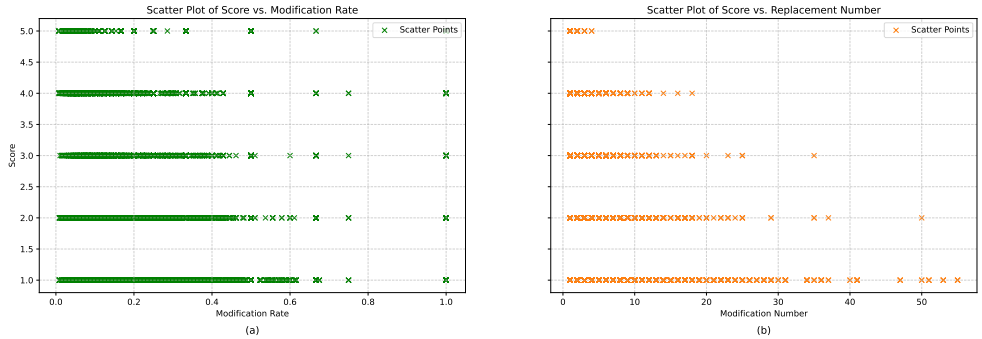


Figure 10: Scatter plots compare Score with modification rate and number for all samples. In subplot (a), the x-axis is the modification rate; in subplot (b), it's the modification number, with the y-axis representing 'Score' in both. Both subplots show a degree of negative correlation between Score and both modification rate and number.

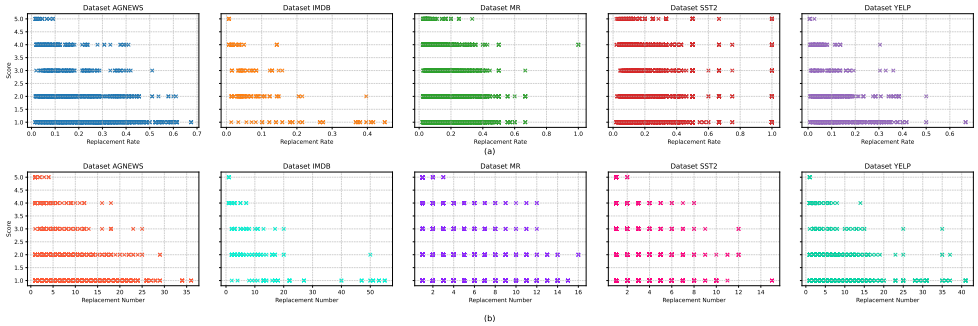


Figure 11: Scatter plots show Score vs. modification rate/number across datasets. X-axis: inversion degree; Y-axis: Score. Part (a): Weak negative Score-modification rate correlation in 'AGNEWS', 'IMDB', 'YELP'; weaker in 'MR'; none in 'SST2.' Part (b): Similar trends for Score-modification number; stronger correlations in 'AGNEWS', 'IMDB', 'YELP'; weaker in 'MR'; none in 'SST2'.

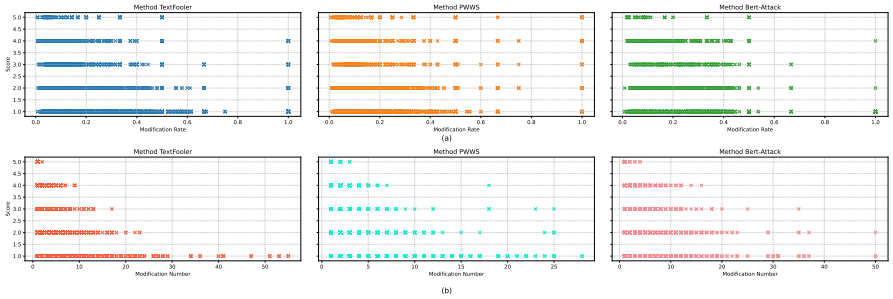


Figure 12: Scatter plots illustrate Score against modification rate and number for different attack methods. The x-axis shows modification rate or number; the y-axis indicates the Score. Part (a) shows Score versus modification rate, revealing no significant correlation across all three attack methods. Part (b) contrasts this, displaying stronger correlations between Score and modification number for the Textfooler and PWWS methods.

inal datasets, and then we generated adversarial examples by different adversarial attacks, we use our fine-tuned tiny-llama to score them, in last, we get a score from 1 to 5 for each samples, and we draw win-rate heatmaps Figure 6 to analyze the performance of different adversarial attacks. We can know that, the BAE is the worst adversarial attack method, and the Faster-Alzantot and A2T are strong adversarial attack methods.

C.2 VScore V.S. # Query

Figures 7 and 8 offer deeper insights into how different attack strategies and datasets impact the query number-quality relationship in adversarial text generation. In addition, Figure 7 shows that VScore falls below 2 when queries exceed 274, highlighting the need to assess query volume's effect on adversarial text generation efficacy. This underscores the importance of detailed analyses across various methods and datasets due to the complexity of adversarial attacks.

Table 10 examines VScore-query number correlations, revealing a nuanced relationship varying by context. In datasets like AGNEWS and IMDB, a strong correlation between VScore and query volume is evident, indicating that query number markedly affects ASR in these cases.

C.3 VScore V.S. Modification Rate or Number

Figure 10 shows the overall relationship between score and modification rate or number. Furthermore, Figure 11 and 12 reveals a notable trend specific to the AGNEWS, IMDB, and YELP datasets. We observe a negative correlation between VScore and both the modification rate and the number of modifications. This indicates that, as the rate of modification or the total number of modifications in adversarial texts increases, the VScore correspondingly decreases.

This trend underscores the necessity of adopting dataset-specific strategies when launching adversarial attacks. The variation in correlation across different datasets suggests that a uniform threshold for modification rate or number may not be universally effective. Instead, attackers should consider setting distinct thresholds tailored to the characteristics and vulnerabilities of each target dataset.

Dataset/Methods		BERTScore		MiniLM		USE		SimCSE	
		Spearman	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman	Pearson
Overall	-	0.2672	0.2701	0.1658	0.1397	0.2346	0.2192	0.2256	0.1677
Datasets	AGNEWS	0.4878	0.4900	0.4392	0.3762	0.4644	0.4145	0.5046	0.4401
	IMDB	0.6874	0.6491	0.3796	0.2896	0.6412	0.5438	0.5743	0.3325
	MR	0.2804	0.3122	0.2448	0.2472	0.2829	0.3053	0.3187	0.2912
	SST2	0.1578	0.1691	0.1498	0.1355	0.1989	0.2029	0.2392	0.2016
	YELP	0.4575	0.4449	0.3796	0.2896	0.4293	0.3456	0.4191	0.2947
Methods	TextFooler	0.3709	0.3643	0.1724	0.1399	0.3106	0.2800	0.2338	0.1794
	PWWS	0.1639	0.1623	0.1007	0.0815	0.1387	0.1119	0.1274	0.0726
	Bert-Attack	0.2208	0.2483	0.2610	0.2629	0.2439	0.2726	0.3676	0.3422

Table 10: Correlation between Score and sentence similarity at different granularities.

C.4 VScore V.S. Sentence Similarity Metrics

We examined metrics like BERTScore, USE, CSESIM, and MiniLM for sentence similarity assessment to gauge their effectiveness in constraining adversarial texts. Analyzing VScore-sentence similarity correlations across granularities (Table 10) revealed generally weak correlations, indicating the limited efficacy of current metrics in capturing the relationship between original and adversarial examples. This suggests a gap in existing metrics' ability to closely measure adversarial texts against originals. Notably, AGNEWS and IMDB showed stronger correlations, highlighting variable metric performance across datasets.

D Human Annotation Details

The annotation is conducted by 5 authors of this paper independently. These annotators are proficient in this domain, with different genders, ages, and educational backgrounds to ensure diversity. They are required to give objective judgments without considering answer lengths to minimize human bias. The prompts for human annotators are the same for GPT-4, which is detailed in Appendix E.

E Prompt Design and Case Study

In Table 11, we detail the dataset construction using a prompt-response format. Figure 13 presents a comprehensive prompt, including one-shot and test samples for reference.

As shown in Figure 14, we present a selection of case studies derived from the results of *AVLLM*, showcasing a range of scores from 1 to 5.

➤ Prompt:	You are a helpful assistant to analysis and rate the Semantic Similarity metric between two text samples: an original text and an adversarial text with changes like synonym substitutions and minor grammatical alterations. Evaluation Criteria: Semantic Similarity (1-5) - This measures how closely the adversarial text aligns with the original in terms of meaning. Changes may include synonym substitutions and minor grammatical errors, highlighted with '[[' and ']]'. The score should reflect the degree to which the altered text preserves the original's meaning, where 5 signifies identical meaning and 1 indicates a completely different meaning. Evaluation Steps: 1. Read both the original and adversarial text samples carefully. 2. Pay attention to the words or sentences modified in the adversarial example, as indicated by '[[' and ']]'. 3. Evaluate the extent to which the adversarial text maintains the semantic essence of the original. 4. Answer by starting to analyze the given example regarding the evaluation criteria as concise as possible (no more than 50 words), and then give the numeric rating. Question: On a scale of 1-5, where 5 is the highest, how similar is the adversarial text to the original in terms of semantic meaning? Your should follow the Evaluation Steps and rate based on Evaluation Criteria. Please adhering to the JSON format as shown in examples. Note that repetition of the provided sentences is unnecessary.
Response:	"Original": "the cast has a high time, but de broca has [[little]] enthusiasm for such antique pulp." "Adversarial": "the cast has a high time, but de broca has [[much]] enthusiasm for such antique pulp." "Analysis": "The change from 'little' to 'much' enthusiasm significantly alters the sentiment about de Broca's attitude towards the 'antique pulp'. This shifts the meaning from negative to positive." "Score": 1.

Table 11: The prompt for GPT-4 to collect our dataset. We give the specific task instructions, steps for chain-of-thought, and one-shot samples.

➤ Analyze two text samples: an original and an adversarial one altered with synonym substitutions and permitted grammatical errors. Your task is to determine if the adversarial example retains the same meaning as the original. We allow some simple syntax errors. Seriously identify the key synonyms and any grammatical deviations used, assess if these changes significantly alter the message or tone, and then score the consistency on a scale from 1 to 5. A score of 5 indicates identical meaning, while 1 suggests a completely different meaning. Conclude with a brief reason for your score.
➤ Examples:
Original: the cast has a high time , but de broca has [[little]] enthusiasm for such antique pulp .
Adversarial: the cast has a high time , but de broca has [[much]] enthusiasm for such antique pulp .
Reason: This change in word choice from "little" to "much" dramatically alters the meaning of the sentence. The structure and the rest of the content remain unchanged, but the sentiment about de Broca's enthusiasm for "such antique pulp" is completely reversed. Therefore, the consistency score is 1 out of 5, reflecting a completely different meaning conveyed by the adversarial example compared to the original.
Score: 1
.....
Original: the [[lady]] and the duke is eric rohmer's economical antidote to the bloated costume drama
Adversarial: the [[woman]] and the duke is eric rohmer's economical antidote to the bloated costume drama
Reason: Despite the slight shift in nuance due to the change from "lady" to "woman," the overall meaning of the sentence is largely unchanged. The core message about Eric Rohmer's film and its contrast to typical costume dramas remains intact. Therefore, the consistency score is 5 out of 5, indicating that the adversarial example retains the same meaning as the original.
Score: 5
➤ Test Sample:
"original": "I just finished watching this movie and am disappointed to say that I didn't enjoy it a bit. It is so [[slow]] Slow and [[uninteresting]]. This kid from Harry Potter [[plays]] a shy teenager with an rude mother, and then one day the rude mother tells the kid to find a job so that they could accommodate an old guy apparently having no place to live has started to live with his family and therefore the kid goes to work for a old lady. And this old lady who is living all alone teaches him about girls, driving car and life! I couldn't get how an 18 year old guy enjoy spending time with an [[awful]] lady in her 80s. Sorry if my comments on this movie has bothered people who might have enjoyed it, I could be wrong as I am not British and may not understand the social and their family structure and way of life. Mostly the movie is made for the British [[audience]].",
"adversarial": "I just finished watching this movie and am disappointed to say that I didn't enjoy it a bit. It is so [[lento]] Slow and [[humdrum]]. This kid from Harry Potter [[dessert]] a shy teenager with an rude mother, and then one day the rude mother tells the kid to find a job so that they could accommodate an old guy apparently having no place to live has started to live with his family and therefore the kid goes to work for a old lady. And this old lady who is living all alone teaches him about girls, driving car and life! I couldn't get how an 18 year old guy enjoy spending time with an [[egregious]] lady in her 80s. Sorry if my comments on this movie has bothered people who might have enjoyed it, I could be wrong as I am not British and may not understand the social and their family structure and way of life. Mostly the movie is made for the British [[listeners]].",

Figure 13: An example of a comprehensive prompt with one-shot samples and input samples.

➤ original text: achieves its main strategic objective
perturbed text: perceives its cardinal political objective
Analysis: The change from 'achieves' to 'perceives' and 'strategic' to 'political' completely alters the context, shifting from accomplishing a goal to merely understanding a central policy.
Score: 1
➤ original text: while undercover brother is definitely one for the masses, it's also full of sharp, smart satire .
perturbed text: while undercover brother is definitely one for the masses, it's too full of sharp, impertinent irony .
Analysis: Changing 'also' to 'too' slightly modifies emphasis, but 'smart satire' to 'impertinent irony' changes the tone from clever to disrespectful, altering the original's complimentary nature.
Score: 2
➤original text: smart and fun , but far more witty than it is wise .
perturbed text: smart and funny , but not more funny than it is wise .
Analysis: Changing 'fun' to 'funny' is minimal, but 'far' to 'not' and repeating 'funny' instead of 'witty' distorts the original comparison. The essence is slightly altered.
Score: 3
➤original text: the film does a solid job of slowly, steadily building up to the climactic burst of violence .
perturbed text: the film does a solid labor of slowly, steadily building up to the climactic burst of violence .
Analysis: Replacing 'job' with 'labor' is a minor alteration; the overall meaning concerning the film's buildup to a key moment of climax is preserved.
Score: 4
➤original text: true star
perturbed text unfeiged star
Analysis: The substitution of 'true' with 'unfeigned' maintains the meaning, as both words convey authenticity or sincerity.
Score: 5

Figure 14: Case study in our proposed dataset. We highlight the modification words in our examples.