Enhancing NLU in Large Language Models Using Adversarial Noisy Instruction Tuning

Shengyuan Bai[1,2*], **Qibin Li**[1*†], **Zhe Wang**[3], **Nai Zhou**[4], **Nianmin Yao**[1†]
[1]School of Computer Science and Technology, Dalian University of Technology
[2]International Digital Economy Academy (IDEA)
[3]Hong Kong University of Science and Technology
[4]Quan Cheng Laboratory
jerry.sy.bai@gmail.com, liqibin@mail.dlut.edu.cn
zwangec@connect.ust.hk, zhounai1992@outlook.com, lucos@dlut.edu.cn

Abstract

Instruction tuning has emerged as an effective approach that notably improves large language models (LLMs) performance, showing particular promise in natural language generation tasks by producing more diverse, coherent, and task-relevant outputs. However, extending instruction tuning to natural language understanding (NLU) tasks presents significant challenges, primarily due to the difficulty in achieving high-precision responses and the scarcity of large-scale, high-quality instruction data necessary for effective tuning. In this work, we introduce *Adversarial Noisy Instruction Tuning* (ANIT) to improve NLU performance on LLMs. First, we leverage low-resource techniques to construct noisy instruction datasets. Second, we employ semantic distortion-aware techniques to quantify the intensity of noise within these instructions. Last, we devise an adversarial training method that incorporates a noise response strategy to achieve noisy instruction tuning. ANIT enhances LLMs capability to detect and accommodate semantic distortions in noisy instructions, thereby augmenting their comprehension of task objectives and ability to generate more accurate responses. We evaluate our approach across diverse noisy instructions and semantic distortion quantification methods on multiple NLU tasks. Comprehensive empirical results demonstrate that our method consistently outperforms existing approaches across various experimental settings.

Introduction

Natural Language Understanding (NLU) and Natural Language Generation (NLG) constitute the two core tasks within the field of Natural Language Processing (NLP). In recent years, large language models (LLMs), exemplified by GPT-4 (OpenAI et al. 2024), Claude-3.5 (Anthropic 2024), and the LLaMA series (Meta 2024), have achieved remarkable breakthroughs in language generation, particularly excelling in producing human-like responses. The success of LLMs can be attributed to the instruction tuning (IT), which enables better alignment with human preferences. However, it is noteworthy that while these models demonstrate impressive performance in NLG tasks, their progress in NLU tasks has been comparatively limited (Li et al. 2023).

Instruction tuning represents the most direct and efficient approach to enhancing the performance of LLMs on specific tasks (Vilar et al. 2022). In the domain of NLU, instruction tuning necessitate high-quality, task-specific instruction data to produce the precise responses required by human. Data augmentation techniques offer a seemingly straightforward solution to expand instruction tuning data. However, previous studies have demonstrated that even minor degradation in data quality or the introduction of noise can lead to significant performance decrements in LLMs (Liu et al. 2023). The construction of specialized instruction tuning datasets for NLU tasks is resource-intensive, both in terms of time and cost, thereby limiting the potential improvements achievable through instruction tuning. This constraint underscores the critical need for developing more efficient and robust method to enhance the NLU performance of LLMs via instruction tuning. Such methods must strike a delicate balance between costs and performance gains, addressing the current limitations.

Recent research has shown that the quality of instructions can be compared using various metrics, such as calculating instruction perplexity and comparing instruction lengths (Zhang et al. 2023). Furthermore, researchers have discovered that fine-tuning LLMs using instructions of varying quality and correcting the output based on quality metrics, noisy instructions can also enhance the performance on downstream tasks (Zhao et al. 2024). However, these approaches do not simultaneously train on data of varying quality levels. Instead, they primarily rely on the mutual correction of different output results, with many methods being non-parametric. Consequently, the generalization capabilities of these techniques are inherently limited.

In pursuit of refining the behavior of LLMs, we have begun to explore the ***Semantic robustness***, a widely recognized cognitive ability in cognitive science (Huang et al. 2021). Semantic robustness refers to the capacity of a language model or natural language processing system to maintain consistent and accurate semantic interpretations across diverse linguistic contexts, variations in input, and potential

[*]These authors contributed equally.
[†]Corresponding author

perturbations. Intriguingly, this cognitive principle has been demonstrated to extend to LLMs. For example, through effective instruction tuning, the outputs generated by LLMs can be steered towards completing specific task rather than understanding the task content. Building upon this foundational insights, we propose the following hypothesis: If LLMs can comprehend multiple noisy instructions and generate excepted outputs consistent with the raw instruction, then the efficacy of instruction tuning for specific tasks within LLMs can be significantly improved.

In this work, we propose *Adversarial Noisy Instruction Tuning* (ANIT), a novel method designed to enhance the performance of LLMs on NLU tasks with minimal additional costs related to instruction augmentation and parameter fine-tuning. The central premise of ANIT involves the deployment of noisy variants of instructions, coupled with the utilization of adversarial training during instruction tuning. Specifically, our approach introduces noise into the raw instructions in a controlled and diverse manner. Moreover, we have developed a new adversarial training strategy, termed the Noise Response Method, which adaptively adjusts the intensity of adversarial perturbations based on the quantification of semantic distortion in the input. Through this precise design, ANIT effectively mitigates the variability of noise in the instructions. This innovative method allows the loss from adversarial training to regularize the original loss function, consequently enhancing the comprehension capabilities of LLMs. Our main contributions are as follows:

- We introduce *Adversarial Noisy Instruction Tuning* (ANIT), a novel method that constructs noisy instructions, quantifies semantic distortion, and utilizes adversarial training in instruction tuning. Through noisy instruction tuning and noise response in adversarial training, LLMs focus on solving NLU tasks without paying attention to irrelevant noisy words of instructions.
- We evaluate the performance of LLMs fine-tuned by ANIT with various noisy instructions across four mainstream NLU tasks. Our results demonstrate that enhancements in semantic robustness directly contribute to performance improvements in downstream tasks. Remarkably, ANIT consistently shows notable performance gains across various models and tasks, regardless of whether the instructions are detailed or concise.
- We provide a comprehensive analysis of ANIT behavior, demonstrating its efficacy from various perspectives. ANIT reduces the high time overhead associated with fine-tuning LLMs using adversarial training. Moreover, the performance gain is maintained in cross-dataset and multitask applications.

Related Work

Instruction Tuning

A considerable corpus of research has illustrated that instruction tuning substantially augments the generalization capabilities of LLMs (Aw et al. 2024). Subsequent to instruction tuning, LLMs exhibit an improved ability to comprehend instructions, leading to notable enhancements in their performance on tuning tasks. Moreover, increasing the diversity and volume of instructions significantly enhances the performance of LLMs (Wang et al. 2023). However, the paucity of high-quality, diverse instructions, heavily dependent on labor-intensive manual annotation, poses a substantial challenge (Yin et al. 2023). Consequently, the prohibitive costs associated with constructing datasets for instruction tuning emerge as a bottleneck in the development of LLMs with enhanced applicability.

Adversarial Training

Adversarial training is widely applied in deep learning, enhancing performance across various domains and tasks. Previous studies categorize adversarial training approaches into two types: single-step (Wong, Rice, and Kolter 2020) and multi-step (Zhang et al. 2019). Multi-step methods, such as PGD (Madry et al. 2018), FreeAT (Shafahi et al. 2019) and FreeLB (Zhu et al. 2020), achieve optimal adversarial perturbations through multiple iterations. However, due to the significant computational power and time required for both pre-training and fine-tuning LLMs, multi-step methods are not directly applicable. Single-step methods, for example, FGSM (Goodfellow, Shlens, and Szegedy 2015) and FGM (Miyato, Dai, and Goodfellow 2017), estimate adversarial perturbations and complete adversarial training in one step. While single-step methods save time and computational resources, they struggle to achieve the optimal effects of adversarial training.

Method

In this section, we present *Adversarial Noisy Instruction Tuning* (ANIT), a method designed to enhance the performance of NLU tasks for LLMs. Figure 1 provides an overview of ANIT. The method has three core parts: (1) Noisy Instruction Construction (2) Semantic Distortion Quantification (3) Noise Response Method for Adversarial Training.

Noisy Instruction Construction

In constructing noisy instructions, we have two primary objectives. First, we aim to create noisy instructions that induce semantic distortion, designed to elicit counter-intuitive yet plausible responses, thereby deviating from expected outcomes. Second, we strive to ensure that the noise within these instructions is controllable. This control prevents high-intensity noise from undermining the effectiveness of instruction tuning and avoids a decline in the performance of LLMs.

Preliminary In our method, an instruction tuning data sample, represented as $X = \{I_r, C\}$, contained two parts: raw instruction I_r and context C. Context C contains the question, format, and option for the specific task (Appendix A). Each sample X corresponds to a target output Y. The noisy instruction represents as I_n, a noisy instruction tuning data sample represented as $X_n = \{I_n, C\}$. In noise instruction tuning, we aim for the LLM outputs of X and X_n to remain as consistent as possible.

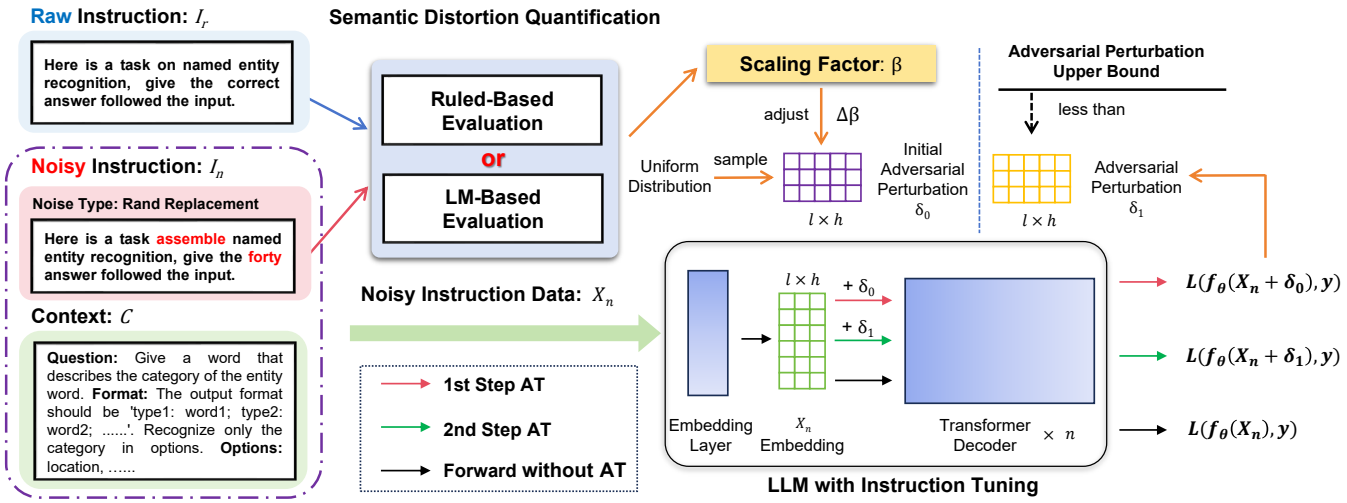

Figure 1: Overview of Adversarial Noisy Instruction Tuning (ANIT). The example in this figure comes from the Conll03 dataset. The raw instruction and corresponding noisy instruction and their semantic distortion quantification are in the left section. The process of how the model performs adversarial training based on these instructions during the training process is in the right section.

Noise in Instructions Our methodology for constructing noisy instructions adheres to three core principles, incorporating six noise variants. These noise variants are elaborated upon in detail in Appendix A.

- **Random Noise:** To minimize the disruption of raw instructions by noisy data, we employed random noise as the foundational method. The construction of random noisy instructions comprises four methods: **random truncation**, **random shuffling**, **random insertion**, and **random replacement**. These methods of introducing randomness reflect common real-world challenges faced by LLMs. Typically, inputs to LLMs are susceptible to errors such as omissions, incorrect words, or altered sequences. The design of random noisy instructions aims to replicate these typical conditions.
- **In-context Noise:** In-context learning enhances the accuracy and relevance of generated outputs by exploiting similarities between inputs and their corresponding pre-trained datasets. For constructing in-context noisy instructions, we utilize a singular method: **in-context padding**. This approach involves embedding extraneous in-context data within raw instructions, aimed at disrupting the generative trajectory of LLMs and thereby introducing controlled noise.
- **Opposite Noise:** Opposite noise is achieved by **instruction confusing** on raw instructions. We address scenarios where instructions are ambiguous or polysemous, resulting in uncertainty for LLMs regarding the exact intent of execution. Such noise enhances the capability to interpret unclear directives of LLMs, discerning task objectives from ambiguity and thereby improving their adaptability.

Noise Strength Adjustment To adjust the strength of noise in the noisy instructions, we introduce a noise strength factor, denoted by ω. This factor is defined as the ratio relative to the length of the raw instruction. A larger value of ω indicates that an increased number of words will undergo replacement, insertion, padding, or other manipulation. Additionally, we established a global noise factor, denoted by ϕ, to determine the proportion of noisy instructions within a dataset. ω and ϕ are both hyperparameters. Details about ω and ϕ is provided in Appendix B.

Semantic Distortion Quantification

To quantify semantic distortion, we employ two distinct approaches: *Rule-Based Evaluation* (**RBE**) and *Language Model-Based Evaluation* (**LME**). RBE is the fuzzy estimate of semantic distortions and LME is the precise estimate of semantic distortions.

Rule-Based Evaluation In RBE, we used the Levenshtein distance algorithm based on a dynamic programming implementation to evaluate the difference between the noisy instructions and raw instructions on the word level (Appendix F). The minimum number of single-word edits obtained according to the Levenshtein distance algorithm represents the distance d between noisy instruction and the raw instruction. For a dataset of noisy instructions, quantification through RBE yields $D = \{d_1, d_2, \ldots, d_n\}$. We applied Max-Min Normalization to this set (Eq. 1).

$$d_i^{'} = \frac{d_i - d_{min}}{d_{max} - d_{min}} \tag{1}$$

For one noisy instruction, we can quantify its semantic distortion as the value $d_i^{'}$. For a noisy instruction dataset, we can obtain the semantic distortion value as β (Eq. 2).

$$\beta = \frac{1}{n} \sum_{i=1}^{n} d_i^{'} \quad d_i^{'} \in D^{'} \tag{2}$$

Model	Method	NER		RE		TC		ABSA	
		ConLL03	Ontonotes	NYT	SciERC	SST2	AGNews	14Lap	14Rest
Gemma-2B	-	91.78	91.03	90.21	38.06	96.73	94.08	62.97	72.03
	Rand Repl	**92.77** ↑	91.47 ↑	**92.00** ↑	**40.05** ↑	97.02 ↑	**95.79** ↑	63.04 ↑	**73.14** ↑
	Rand Trunc	92.32 ↑	**91.83** ↑	91.25 ↑	39.51 ↑	96.61	94.63 ↑	62.94 ↑	72.71 ↑
	Rand Ins	92.53 ↑	91.39 ↑	91.93 ↑	38.88 ↑	97.11 ↑	95.75 ↑	63.01 ↑	72.50 ↑
	Rand Shuf	92.44 ↑	91.66 ↑	91.81 ↑	39.04 ↑	96.91 ↑	95.43 ↑	63.08 ↑	72.97 ↑
	IC Pad	91.61 ↑	91.61	91.07 ↑	39.73 ↑	**97.12** ↑	95.77 ↑	**63.17** ↑	73.04 ↑
	Opposite	92.07 ↑	91.87 ↑	91.22 ↑	39.29 ↑	96.88 ↑	95.05 ↑	62.77	72.98 ↑
LLaMA2-7B	-	92.63	90.39	90.89	42.18	96.80	94.08	62.97	72.37
	Rand Repl	**93.64** ↑	91.94 ↑	**92.74** ↑	**45.36** ↑	97.18 ↑	**95.88** ↑	**64.72** ↑	**74.21** ↑
	Rand Trunc	93.01 ↑	91.89 ↑	92.21 ↑	44.50 ↑	97.01 ↑	94.58 ↑	64.07 ↑	74.00 ↑
	Rand Ins	93.26 ↑	91.33 ↑	92.53 ↑	43.50 ↑	97.37 ↑	94.75 ↑	63.50 ↑	73.75 ↑
	Rand Shuf	93.33 ↑	91.26 ↑	92.60 ↑	44.21 ↑	97.13 ↑	95.47 ↑	63.98 ↑	74.15 ↑
	IC Pad	93.45 ↑	91.78 ↑	92.47 ↑	45.14 ↑	**97.19** ↑	95.73 ↑	64.53 ↑	73.51 ↑
	Opposite	93.17 ↑	**91.97** ↑	91.79 ↑	42.90 ↑	96.85 ↑	94.95 ↑	63.75 ↑	73.50 ↑
LLaMA3-8B	-	92.71	91.53	91.57	48.84	97.40	95.02	64.37	73.84
	Rand Repl	**93.66** ↑	92.37 ↑	**92.94** ↑	**50.57** ↑	97.56 ↑	**95.89** ↑	**66.13** ↑	**74.93** ↑
	Rand Trunc	93.31 ↑	92.05 ↑	92.55 ↑	50.13 ↑	**97.81** ↑	94.87	65.10 ↑	74.72 ↑
	Rand Ins	93.58 ↑	**92.42** ↑	92.75 ↑	49.53 ↑	97.61 ↑	94.99	65.47 ↑	74.15 ↑
	Rand Shuf	93.03 ↑	91.37	92.71 ↑	49.21 ↑	97.56 ↑	95.88 ↑	65.48 ↑	74.34 ↑
	IC Pad	93.05 ↑	92.17 ↑	92.61 ↑	49.84 ↑	97.25	95.74 ↑	64.13	74.07 ↑
	Opposite	93.51 ↑	92.23 ↑	92.04 ↑	49.59 ↑	97.11	95.26 ↑	65.59 ↑	74.11 ↑

Table 1: F1 Scores across NLU tasks: Performance of ANIT with detailed instructions on LLMs, semantic distortion assessed through rule-based evaluation. Best results are highlighted in bold; performance improvements are indicated by ↑; fine-tuned LLMs without ANIT as the baseline are indicated in underline.

The β is used as the scaling factor to regulate the initial perturbation strength of adversarial training. A larger value of β indicates increased semantic distortion and vice verse.

LM-Based Evaluation In LME, we used OpenAI API with `text-embedding-3-large`[1] as the embedding model. `text-embedding-3-large` is a LM developed to measure text embedding similarity, and it is one of the best text embedding models currently available. We used `text-embedding-3-large` for embedding of noisy instructions and raw instructions as different inputs to get two embedding matrix. Finally, the cosine similarity algorithm is used to measure the semantic distortion of the two output embedding matrices. By representing the embedding model `text-embedding-3-large` as $E(x)$, semantic distortion on each noisy instruction can be quantified by $d_i^{'} = 1 - cos(E(x_{in}), E(x_i))$, the scaling factor β can be derived (Eq. 3).

$$\beta = \frac{1}{n}\sum(1 - cos(E(x_{in}), E(x_i))) \quad x_{in} \in X_n, x_i \in X \tag{3}$$

Noise Response Method for Adversarial Training

The core idea of adversarial training (AT) is the process of modifying the training objectives by applying perturbation δ to the input and maximizing the adversarial loss (Eq. 4). Specifically, AT aims to find the most appropriate parameters θ to find the maximum disturbance δ within the standard norm ball and minimize the standard error with the output. D is the data distribution, y is the label, ϵ is perturbation bound and L is the loss function.

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim D}\left[\max_{\|\delta\|\leq\epsilon} L(f_\theta(X+\delta), y)\right] \tag{4}$$

The outer minimization problem can be solved by gradient descent. The inner maximization problem can be solved by running projected gradient descent on several negative loss functions. Specifically, the following steps are taken in each iteration (with a step size of α):

$$g_{adv} \leftarrow \nabla_\delta L(f_\theta(X+\delta_i), y) \tag{5}$$

$$\delta_{i+1} = \Pi_{||\delta||\leq\epsilon}(\delta_i + \alpha \cdot g_{adv} / ||g_{adv}||_F) \tag{6}$$

where Eq. 5 is the gradient of the loss at the i-th step with respect to δ_i, $||g_{adv}||_F$ is the F-norm of the gradient, and $\Pi_{||\delta||\leq\epsilon}$ is the projection of the performance onto the perturbation bound.

Traditional AT methods, such as the Fast Gradient Method (FGM) (Miyato, Dai, and Goodfellow 2017) and Free Large-Batch (FreeLB) (Zhu et al. 2020), are the straightforward techniques that improves the performance of neural networks. However, FGM as a single-step adversarial training approach often falls short in precision of adversarial perturbation estimation, leading to sub-optimal performance on LMs compared to FreeLB. While FreeLB is effective for BERT-based language models (Devlin et al. 2019), it requires multiple adjustments of adversarial perturbations within a batch, resulting in significant additional training overhead that limits its widespread applicability in LLMs.

[1]https://platform.openai.com/docs/guides/embeddings

Model	Method	NER		RE		TC		ABSA	
		ConLL03	Ontonotes	NYT	SciERC	SST2	AGNews	14Lap	14Rest
Gemma-2B	-	91.59	90.87	90.09	37.74	96.78	93.98	62.64	71.73
	Rand Repl	**92.24** ↑	91.33 ↑	**91.86** ↑	**39.46** ↑	96.90 ↑	**95.68** ↑	**62.98** ↑	**72.74** ↑
	Rand Trunc	92.14 ↑	**91.79** ↑	90.96 ↑	39.27 ↑	96.74	94.33 ↑	62.53	72.34 ↑
	Rand Ins	92.23 ↑	91.01 ↑	91.71 ↑	38.83 ↑	**97.04** ↑	95.26 ↑	62.79 ↑	72.15 ↑
	Rand Shuf	91.97 ↑	91.23 ↑	91.64 ↑	39.14 ↑	96.85 ↑	95.31 ↑	62.83 ↑	72.59 ↑
	IC Pad	91.96 ↑	91.74 ↑	89.74 ↑	39.33 ↑	96.93 ↑	95.46 ↑	62.94 ↑	72.48 ↑
	Opposite	91.32	91.78 ↑	90.97 ↑	38.29 ↑	96.59 ↑	94.79 ↑	62.37	72.65 ↑
LLaMA2-7B	-	92.35	90.06	90.56	41.80	96.67	93.82	62.68	71.17
	Rand Repl	**93.10** ↑	**91.95** ↑	**92.18** ↑	**44.40** ↑	**97.24** ↑	**95.64** ↑	64.26 ↑	74.25 ↑
	Rand Trunc	92.86 ↑	91.78 ↑	92.03 ↑	43.69 ↑	97.17 ↑	94.36 ↑	63.98 ↑	**74.37** ↑
	Rand Ins	92.95 ↑	91.35 ↑	92.26 ↑	44.18 ↑	96.99 ↑	95.27 ↑	**65.14** ↑	73.78 ↑
	Rand Shuf	93.19 ↑	91.76 ↑	92.01 ↑	43.91 ↑	97.06 ↑	94.75 ↑	64.27 ↑	73.93 ↑
	IC Pad	92.73 ↑	91.13 ↑	92.15 ↑	43.75 ↑	96.83 ↑	94.81 ↑	64.68 ↑	73.59 ↑
	Opposite	92.73 ↑	91.88 ↑	91.76 ↑	42.67 ↑	96.72 ↑	94.05 ↑	63.86 ↑	74.13 ↑
LLaMA3-8B	-	92.48	91.46	91.23	48.17	97.01	94.64	64.49	73.88
	Rand Repl	**93.31** ↑	**92.04** ↑	**92.78** ↑	**50.00** ↑	97.48 ↑	95.78 ↑	**65.89** ↑	**74.59** ↑
	Rand Trunc	93.03 ↑	91.72 ↑	92.35 ↑	49.76 ↑	**97.77** ↑	94.88 ↑	64.55	74.39 ↑
	Rand Ins	93.21 ↑	92.00 ↑	92.64 ↑	49.41 ↑	97.47 ↑	94.23 ↑	65.07 ↑	73.94 ↑
	Rand Shuf	93.13 ↑	91.93 ↑	92.59 ↑	49.19 ↑	97.26 ↑	**95.88** ↑	64.04	74.27 ↑
	IC Pad	92.84 ↑	91.29	92.77 ↑	49.30 ↑	96.85	95.54 ↑	65.45 ↑	73.56
	Opposite	92.17	91.71 ↑	91.91 ↑	49.48 ↑	97.64 ↑	94.97 ↑	65.19 ↑	74.05 ↑

Table 2: F1 Scores across NLU tasks: Performance of ANIT with concise instructions on LLMs, semantic distortion assessed through rule-based evaluation. Best results are highlighted in bold; performance improvements are indicated by ↑; fine-tuned LLMs without ANIT as the baseline are indicated in underline.

We propose a novel adversarial training method, the *Noise Response Method* (NRM). NRM can accomplish a dynamic adversarial training that requires only two steps based on the semantic distortion quantification value. NRM adaptively initializes adversarial perturbations based on the scaling factor β derived from the quantified semantic distortion of the input, achieving adversarial training in just two steps.

$$\delta_0 \leftarrow \frac{10}{\sqrt{N_\delta}} U(-\epsilon, \epsilon) \cdot (1-\beta) \tag{7}$$

Algorithm 1: Noise Response Method for AT

Require: Noisy instruction tuning sample $X_n = (I_n, C)$, perturbation bound ϵ, learning rate τ, adversarial coefficient α, scaling coefficient β

Initialize LLM parameters θ
for epoch $= 1 \ldots N$ **do**
  Semantic distortion evaluation:
  $\beta = RBE(X_n, X)$ or $LME(X_n, X)$
  for minibatch $B \subset X_n$ **do**
    $\delta_0 \leftarrow \frac{10}{\sqrt{N_\delta}} U(-\epsilon, \epsilon) \cdot (1-\beta)$
    $g_{adv} \leftarrow \nabla_\delta L(f_\theta(X_n + \delta_0), y)$
    $\delta_1 = \Pi_{||\delta|| \le \epsilon}(\delta_0 + \alpha \cdot g_{adv} / ||g_{adv}||_F)$
  end for
  $L = L(f_\theta(X_n), y) + \lambda L(f_\theta(X_n + \delta_1), y)$
  $g_\theta = \nabla_\theta L$
  $\theta \leftarrow \theta - \tau g_\theta$
end for

In the first step of the adversarial process, δ_0 is sampled from a uniform distribution $U(-\epsilon, \epsilon)$ and subsequently scaled by a factor β (Eq. 7), where N_δ denotes the dimension of the δ, $\frac{10}{\sqrt{N_\delta}}$ and $(1-\beta)$ together form a scaling factor. This scaling mechanism modulates the intensity of AT: it reduces the training intensity when the noise is excessive and increases it when the noise is minimal. Such adaptability ensures the effectiveness of adversarial training and enhances the robustness of LLMs to various noise strength during instruction tuning.

In the second step of the adversarial process, we enforce a constraint ensuring that the perturbation δ_1 does not exceed the upper bound, denoted by ϵ. This constraint is critical for the effectiveness of AT (Goodfellow, Shlens, and Szegedy 2014). The hyperparameter α, which acts as the adversarial coefficient, is employed to regulate the extent of the perturbation applied during training.

In the loss function (Eq. 8), the loss from AT is added as a regularization term to the standard loss for instruction tuning. λ is a hyperparameter to adjust the loss.

$$L = L(f_\theta(X_n), y) + \lambda L(f_\theta(X_n + \delta_1), y) \tag{8}$$

Experimental Setup

Datasets

We conduct experiments on four representative NLU tasks: Named Entity Recognition (NER), Relationship Extraction (RE), Text Classification (TC) and Aspect-based Sentiment Analysis (ABSA). For each task, we employ two datasets: Ontonotes (Hovy et al. 2006) and CoNLL2003 (Tjong

Model	SD Evaluation Method	NER		RE		TC		ABSA	
		ConLL03	Ontonotes	NYT	SciERC	SST2	AGNews	14Lap	14Rest
Gemma-2B	-	91.78	91.03	90.21	38.06	96.73	94.08	62.97	72.03
	Rule-Based	92.77(+0.99)	91.83(+0.80)	92.00(+1.79)	40.05(+1.99)	97.12(+0.39)	95.79(+1.71)	63.17(+0.20)	73.14(+1.11)
	LM-Based	**92.88**(+1.10)	**91.98**(+0.95)	**92.12**(+1.91)	**40.25**(+2.19)	**97.21**(+0.48)	**96.13**(+2.05)	63.14(+0.17)	**73.40**(+1.37)
LLaMA2-7B	-	92.63	90.39	90.89	42.18	96.80	94.08	62.97	72.37
	Rule-Based	93.64(+1.01)	91.97(+1.58)	92.74(+1.85)	45.36(+3.18)	97.19(+0.39)	95.88(+1.80)	64.72(+1.75)	74.21(+1.84)
	LM-Based	**93.94**(+1.31)	**92.03**(+1.64)	**92.92**(+2.03)	**45.53**(+3.35)	**97.49**(+0.69)	**96.02**(+1.94)	64.69(+1.72)	**74.33**(+1.96)
LLaMA3-8B	-	92.71	91.53	91.57	48.84	97.40	95.02	64.37	73.84
	Rule-Based	93.66(+0.95)	92.42(+0.89)	92.94(+1.37)	50.57(+1.73)	97.81(+0.41)	95.69(+0.67)	66.13(+1.76)	74.93(+1.09)
	LM-Based	**93.79**(+1.08)	**92.82**(+1.29)	**93.58**(+2.01)	**50.66**(+1.82)	**98.16**(+0.76)	**95.87**(+0.85)	**66.16**(+1.79)	**75.26**(+1.42)

Table 3: F1 Scores for NLU Tasks: The comparative results of ANIT performance with detailed instructions on LLMs. This comparison includes different *Semantic Distortion* (SD) Evaluation Methods: Rule-Based and LM-Based. Performance improvements are indicated in parenthesis, fine-tuned LLMs without ANIT as the baseline are indicated in underline.

Kim Sang and De Meulder 2003) for NER; SciERC (Luan et al. 2018) and NYT (Riedel, Yao, and McCallum 2010) for RE; SST2 (Socher et al. 2013) and AGNews (Zhang, Zhao, and LeCun 2015) for TC; 14Lap and 14Rest (Xu et al. 2020) for ABSA.

We collect instructions for each dataset from Alpaca (Taori et al. 2023). Instructions for each task are presented in two versions: **concise** and **detailed**. The concise instructions contain only the target of the task. The detailed instructions provide step-by-step guidance for completing the task. Examples prompts for each task are presented in Appendix A.

Models

In our study, we use Gemma-2B, LLaMA2-7B and LLaMA3-8B for our experiments. These models cover the parameter ranges commonly employed in LLMs. LLaMA3 represents an improvement over LLaMA2, which is achieved through the use of expanded pre-training data and an augmented vocabulary. In our experiments, we employ greedy decoding for these models.

Evaluation Metrics

We examine the output of LLMs by employing ANIT on NLU tasks. Task performance is measured using using Micro-F1 (Manning, Raghavan, and Schütze 2008). We employ different evaluation settings on generated tokens for each task. we prove evaluation details in Appendix C.

Fine-tuning Experimental Setup

To evaluate the effectiveness of ANIT on all models, we fine-tune models using LoRA (Hu et al. 2022), a parameter-efficient fine-tuning method. We conduct experiments both on concise instructions and detailed instructions. We prove implementation details in Appendix C.

Results

ANIT on Detailed Instructions

Table 1 presents the results of applying *Adversarial Noisy Instruction tuning* (ANIT) to LLMs on detailed instructions. The results demonstrate that the application of ANIT consistently enhances the performance of LLMs across diverse NLU tasks, surpassing the baseline that solely employ standard instructions. In the majority of cases, the introduction of different noise yields performance improvements, with the 'Rand Replacement' noise proving to be the most effective. Few noise lead to performance degradation on specific tasks with detailed instructions, indicating the existence of potential limitations in fine-tuning. This phenomenon may be attributed to the pre-training strategies of specific LLM and the construction of instructions.

ANIT on Concise Instructions

Table 2 presents an extended analysis of the efficacy of ANIT in LLMs applied to concise instructions. The inherent brevity and limited information content of concise instructions tend to exacerbate semantic distortions when noise is introduced during instruction tuning. Despite these challenges, we observed that ANIT with concise instructions continues to improve the performance of LLMs on NLU tasks. In contrast to the complexity of detailed instructions, the semantic distortions arising from concise instructions are more straightforward, eliciting a more direct adversarial effect. Consequently, a portion of the noise that demonstrate sub-optimal performance with detailed instructions exhibit improvement when fine-tuning with concise instructions.

Analyzing Different Evaluation Methods to Semantic Distortion

When employing RBE to quantify semantic distortion in ANIT, we observe substantial performance improvements across various NLU tasks using LLMs in Table 1 and 2. This suggests that rule-based evaluation effectively captures semantic distortion, leading to more refined models. To further investigate the impact of semantic distortion quantification on ANIT performance, We select the best-performing noise configurations obtained from ANIT using various LLMs and conduct experiments using the LM-based evaluation. By focusing on the noise configurations that yield the highest performance gains in ANIT, we can better assess the effectiveness of the LM-based evaluation. As shown in Table 3, our findings indicate that the LM-based evaluation results

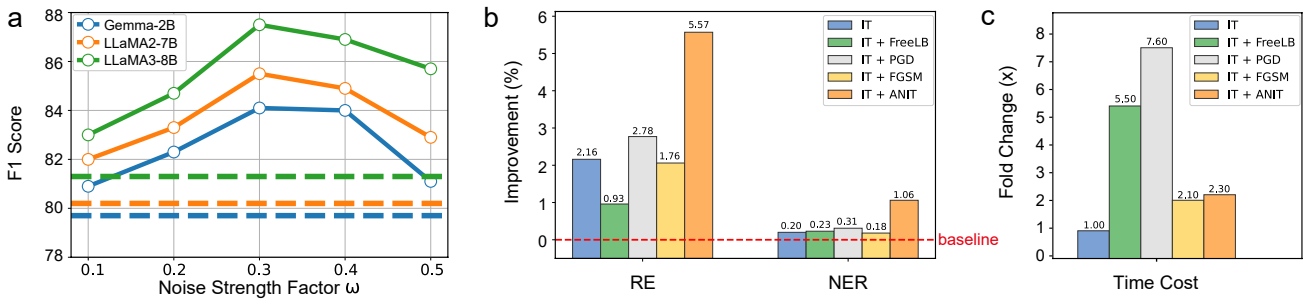

Figure 2: (a): Average F1 scores for various noise strength factors ω are calculated for each LLM fine-tuned with ANIT and evaluated on four NLU tasks. The horizontal lines, color-coded to match each LLM, indicate baseline performance. (b): Time cost and task performance comparison between FreeLB, PGD, FGSM, and ANIT. ANIT uses 'Random Replacement' noise, while others show results from the best 5-round adversarial training. (c): Time-performance comparison between ANIT and other methods.

in greater performance gains. This suggests that LM-based evaluation is more effective in quantifying semantic distortion and enhancing the robustness of LLMs fine-tuning using ANIT for NLU tasks.

Ablation Study

Generalization Capabilities of ANIT

To evaluate the adaptability and effectiveness of the fine-tuned LLMs by ANIT, we conducted a cross-evaluation wherein an LLM trained on one task was subsequently trained and tested with a limited number of samples on another task. Specifically, a model trained on the CoNLL2003 dataset was retrained and tested with few samples from the Ontonotes dataset. The results, detailed in Table 4, show the F1 scores on these tasks using LLaMA2-7B. Across various few-shot training data scenarios, the model consistently outperforms the baseline. Notably, when using only 5% of the training data, the LLM demonstrates an average performance improvement of 2.47%. These findings suggest that ANIT enhances both the generalization ability and robustness of LLMs.

Sensitivity of Noisy Factor in Instructions

Figure 2a illustrates the impact of the noise strength factor ω on our method's performance. This parameter adjusts the semantic distortion caused by noisy instructions in ANIT. While we typically use ω values between 0.1 and 0.3, we assessed ANIT with ω ranging from 0.1 to 0.5 in 0.1 increments. Using the top-performing noise configurations, we calculated the average F1 score across all tasks. Performance declines when ω exceeds 0.3 but remains above the baseline at $\omega = 0.1$. Overall, ANIT enhances model performance within the ω range of 0.1 to 0.5, showing strong stability between 0.1 and 0.3.

On the Utility of ANIT over Classic Adversarial Training

In the ANIT method, we introduce the Noise Response Method (NRM), a novel adversarial training methodology which employs a two-step adversarial process. Compared to direct instruction tuning of LLMs, NRM indeed incurs additional time costs. To comprehensively assess these costs and their impact on performance, We compare NRM with common adversarial training methods such as FreeLB, PGD, and FGSM. As shown in Figure 2b, 2c, although ANIT introduces additional time costs, it is more time-efficient than the other methods and significantly outperforms them in terms of performance. This finding supports our hypothesis that traditional adversarial training methods often struggle to improve model performance across varied LLM applications.

Model	ConLL03	Ontonotes
Baseline		
1%	78.74	74.31
5%	86.98	78.69
10%	87.40	82.75
ANIT		
1%	81.32	75.84
5%	89.26	81.43
10%	90.45	84.67

Table 4: F1 scores obtained by cross-validating the performance of LLaMA2-7B on two datasets.

Conclusion

In this paper, we investigate how to improve the performance of LLMs on NLU tasks by enhancing their semantic robustness without the need for annotating high-quality instructions. We propose the *Adversarial Noisy Instruction Tuning* (ANIT), which significantly improves the ability of LLMs to process complex context. Through extensive analyses, we find that the performance gains obtained using ANIT can be maintained across datasets and tasks. This provides new insights and methods for the application of LLMs in various downstream tasks.

Acknowledgments

This work was supported by by Science and Technology Innovation Key R&D Program of Chongqing, CSTB2024TIAD-STX0027.

References

Anthropic. 2024. Claude 3 haiku: our fastest model yet.

Aw, K. L.; Montariol, S.; AlKhamissi, B.; Schrimpf, M.; and Bosselut, A. 2024. Instruction-tuned LLMs with World Knowledge are More Aligned to the Human Brain.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Hovy, E.; Marcus, M.; Palmer, M.; Ramshaw, L.; and Weischedel, R. 2006. OntoNotes: The 90% Solution. In Moore, R. C.; Bilmes, J.; Chu-Carroll, J.; and Sanderson, M., eds., *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, 57–60. New York City, USA: Association for Computational Linguistics.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Huang, S.; Li, Z.; Qu, L.; and Pan, L. 2021. On Robustness of Neural Semantic Parsers. In Merlo, P.; Tiedemann, J.; and Tsarfaty, R., eds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 3333–3342. Online: Association for Computational Linguistics.

Li, Z.; Li, X.; Liu, Y.; Xie, H.; Li, J.; Wang, F.-l.; Li, Q.; and Zhong, X. 2023. Label supervised llama finetuning. *arXiv preprint arXiv:2310.01208*.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2023. GPT understands, too. *AI Open*.

Luan, Y.; He, L.; Ostendorf, M.; and Hajishirzi, H. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3219–3232. Brussels, Belgium: Association for Computational Linguistics.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Meta. 2024. Llama 3. .

Miyato, T.; Dai, A. M.; and Goodfellow, I. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. In *International Conference on Learning Representations*.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; Avila, R.; Babuschkin, I.; Balaji, S.; Balcom, V.; Baltescu, P.; Bao, H.; Bavarian, M.; Belgum, J.; Bello, I.; Berdine, J.; Bernadett-Shapiro, G.; Berner, C.; Bogdonoff, L.; Boiko, O.; Boyd, M.; Brakman, A.-L.; Brockman, G.; Brooks, T.; Brundage, M.; Button, K.; Cai, T.; Campbell, R.; Cann, A.; Carey, B.; Carlson, C.; Carmichael, R.; Chan, B.; Chang, C.; Chantzis, F.; Chen, D.; Chen, S.; Chen, R.; Chen, J.; Chen, M.; Chess, B.; Cho, C.; Chu, C.; Chung, H. W.; Cummings, D.; Currier, J.; Dai, Y.; Decareaux, C.; Degry, T.; Deutsch, N.; Deville, D.; Dhar, A.; Dohan, D.; Dowling, S.; Dunning, S.; Ecoffet, A.; Eleti, A.; Eloundou, T.; Farhi, D.; Fedus, L.; Felix, N.; Fishman, S. P.; Forte, J.; Fulford, I.; Gao, L.; Georges, E.; Gibson, C.; Goel, V.; Gogineni, T.; Goh, G.; Gontijo-Lopes, R.; Gordon, J.; Grafstein, M.; Gray, S.; Greene, R.; Gross, J.; Gu, S. S.; Guo, Y.; Hallacy, C.; Han, J.; Harris, J.; He, Y.; Heaton, M.; Heidecke, J.; Hesse, C.; Hickey, A.; Hickey, W.; Hoeschele, P.; Houghton, B.; Hsu, K.; Hu, S.; Hu, X.; Huizinga, J.; Jain, S.; Jain, S.; Jang, J.; Jiang, A.; Jiang, R.; Jin, H.; Jin, D.; Jomoto, S.; Jonn, B.; Jun, H.; Kaftan, T.; Łukasz Kaiser; Kamali, A.; Kanitscheider, I.; Keskar, N. S.; Khan, T.; Kilpatrick, L.; Kim, J. W.; Kim, C.; Kim, Y.; Kirchner, J. H.; Kiros, J.; Knight, M.; Kokotajlo, D.; Łukasz Kondraciuk; Kondrich, A.; Konstantinidis, A.; Kosic, K.; Krueger, G.; Kuo, V.; Lampe, M.; Lan, I.; Lee, T.; Leike, J.; Leung, J.; Levy, D.; Li, C. M.; Lim, R.; Lin, M.; Lin, S.; Litwin, M.; Lopez, T.; Lowe, R.; Lue, P.; Makanju, A.; Malfacini, K.; Manning, S.; Markov, T.; Markovski, Y.; Martin, B.; Mayer, K.; Mayne, A.; McGrew, B.; McKinney, S. M.; McLeavey, C.; McMillan, P.; McNeil, J.; Medina, D.; Mehta, A.; Menick, J.; Metz, L.; Mishchenko, A.; Mishkin, P.; Monaco, V.; Morikawa, E.; Mossing, D.; Mu, T.; Murati, M.; Murk, O.; Mély, D.; Nair, A.; Nakano, R.; Nayak, R.; Neelakantan, A.; Ngo, R.; Noh, H.; Ouyang, L.; O'Keefe, C.; Pachocki, J.; Paino, A.; Palermo, J.; Pantuliano, A.; Parascandolo, G.; Parish, J.; Parparita, E.; Passos, A.; Pavlov, M.; Peng, A.; Perelman, A.; de Avila Belbute Peres, F.; Petrov, M.; de Oliveira Pinto, H. P.; Michael; Pokorny; Pokrass, M.; Pong, V. H.; Powell, T.; Power, A.; Power, B.; Proehl, E.; Puri, R.; Radford, A.; Rae, J.; Ramesh, A.; Raymond, C.; Real, F.; Rimbach, K.; Ross, C.; Rotsted, B.; Roussez, H.; Ryder, N.; Saltarelli, M.; Sanders, T.; Santurkar, S.; Sastry, G.; Schmidt, H.; Schnurr, D.; Schulman, J.; Selsam, D.; Sheppard, K.; Sherbakov, T.; Shieh, J.; Shoker, S.; Shyam, P.; Sidor, S.; Sigler, E.; Simens, M.; Sitkin, J.; Slama, K.; Sohl, I.; Sokolowsky, B.; Song, Y.; Staudacher, N.; Such, F. P.; Summers, N.; Sutskever, I.; Tang, J.; Tezak, N.; Thompson, M. B.; Tillet, P.; Tootoonchian, A.; Tseng, E.; Tuggle, P.; Turley, N.; Tworek, J.; Uribe, J. F. C.; Vallone, A.; Vijayvergiya, A.; Voss, C.; Wainwright, C.; Wang, J. J.; Wang, A.; Wang, B.; Ward, J.; Wei, J.; Weinmann, C.; Welihinda, A.; Welinder, P.; Weng, J.; Weng, L.; Wiethoff, M.; Willner, D.; Winter, C.; Wolrich, S.; Wong, H.; Workman, L.; Wu, S.; Wu, J.; Wu, M.; Xiao, K.; Xu, T.; Yoo, S.; Yu, K.; Yuan, Q.; Zaremba, W.; Zellers, R.; Zhang, C.; Zhang, M.; Zhao, S.; Zheng, T.; Zhuang, J.; Zhuk, W.; and Zoph, B. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling Relations and Their Mentions without Labeled Text. In Balcázar, J. L.; Bonchi, F.; Gionis, A.; and Sebag, M., eds., *Machine Learning and Knowledge Discovery in Databases*, 148–163. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-15939-8.

Shafahi, A.; Najibi, M.; Ghiasi, M. A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Yarowsky, D.; Baldwin, T.; Korhonen, A.; Livescu, K.; and Bethard, S., eds., *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–1642. Seattle, Washington, USA: Association for Computational Linguistics.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model.

Tjong Kim Sang, E. F.; and De Meulder, F. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 142–147.

Vilar, D.; Freitag, M.; Cherry, C.; Luo, J.; Ratnakar, V.; and Foster, G. 2022. Prompting palm for translation: Assessing strategies and performance. *arXiv preprint arXiv:2211.09102*.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 13484–13508. Toronto, Canada: Association for Computational Linguistics.

Wong, E.; Rice, L.; and Kolter, J. Z. 2020. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*.

Xu, L.; Li, H.; Lu, W.; and Bing, L. 2020. Position-Aware Tagging for Aspect Sentiment Triplet Extraction. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2339–2349. Online: Association for Computational Linguistics.

Yin, F.; Vig, J.; Laban, P.; Joty, S.; Xiong, C.; and Wu, C.-S. 2023. Did You Read the Instructions? Rethinking the Effectiveness of Task Definitions in Instruction Learning. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3063–3079. Toronto, Canada: Association for Computational Linguistics.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, 7472–7482. PMLR.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level Convolutional Networks for Text Classification. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Zhang, Z.; Wang, S.; Yu, W.; Xu, Y.; Iter, D.; Zeng, Q.; Liu, Y.; Zhu, C.; and Jiang, M. 2023. Auto-Instruct: Automatic Instruction Generation and Ranking for Black-Box Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 9850–9867. Singapore: Association for Computational Linguistics.

Zhao, Y.; Yan, L.; Sun, W.; Xing, G.; Wang, S.; Meng, C.; Cheng, Z.; Ren, Z.; and Yin, D. 2024. Improving the Robustness of Large Language Models via Consistency Alignment. arXiv:2403.14221.

Zhu, C.; Cheng, Y.; Gan, Z.; Sun, S.; Goldstein, T.; and Liu, J. 2020. FreeLB: Enhanced Adversarial Training for Natural Language Understanding. In *International Conference on Learning Representations*.