StanceAttack: Adversarial Attack for Stance Detection

Chenye Zhao **Cornelia Caragea**
Computer Science
University of Illinois Chicago
czhao43@uic.edu cornelia@uic.edu

Abstract

Stance detection aims to ascertain whether an author's text is in favor, against, or neutral toward specific targets like *public policies* or *social issues*. While pretrained language models (PLMs) have greatly enhanced stance detection, they remain vulnerable to adversarial attacks—manipulations that maintain textual semantics but lead to incorrect predictions. Such vulnerabilities remain underexplored for stance detection. In this study, we introduce *StanceAttack*, an innovative adversarial attack method leveraging ChatGPT to create adversarial examples that can mislead well-trained stance detection models. We conduct experiments to evaluate our attack model by attacking state-of-the-art PLMs on two benchmark datasets. Results demonstrate that *StanceAttack* outperforms traditional adversarial methods with higher success rates and fewer retries. Human evaluations confirm that our adversarial examples preserve the original semantic meanings and naturalness. We share our code and data in https://github.com/chenyez/StanceAttack.

1 Introduction

The goal of stance detection is to automatically detect whether the author of a text is in favor, against, or neutral toward a specific (potentially controversial) target (Mohammad et al., 2016; Küçük and Can, 2020; ALDayel and Magdy, 2021), e.g., "private education". The stance can reveal valuable insights relevant to important events such as public policy making and presidential elections by enabling authorities and political entities to understand and address public opinions on issues like health guidelines and education. This capability is crucial for adapting to rapidly changing public sentiments, thereby supporting informed decision-making and enhancing democratic processes (Li et al., 2021; Glandt et al., 2021). With the recent success of works on stance detection, pretrained language models (PLMs) have been widely used to develop strong stance detection classifiers, achieving remarkable performance and currently outperforming large language models (LLMs) on stance detection benchmarks (see Appendix A). However, due to its high-stakes (e.g., changing the outcome of events like presidential elections in polls or changing the outcome of a survey for policy making), the task is prone to context-aware adversarial attacks.

Recent research in NLP has demonstrated that even the most advanced PLMs are susceptible to carefully crafted adversarial examples (Wang et al., 2021). These adversarial examples preserve the semantic meanings of the original input but can deceive PLMs into making incorrect predictions. Exploring adversarial attack methods for stance detection is crucial to understanding the reliability and robustness of PLMs. It is also crucial to develop effective attack algorithms to pinpoint the weaknesses in PLMs before they are deployed, which is the first step toward enhancing their robustness (Wang et al., 2022). In the context of stance detection, adversarial attacks can result in serious consequences, such as political manipulation. While previous research has started to explore adversarial evaluation by testing models against unseen targets or investigating "blind attacks" (Allaway et al., 2021; Conforti et al., 2021; Chunling et al., 2023; Schiller et al., 2021), the development of adversarial attack algorithms specifically designed to generate examples that compromise stance detection models remains unexplored.

Existing word-level adversarial attacks involve perturbing sentences by inserting new words, removing words, or changing words. These types of attacks generally achieve higher success rates than character-level or sentence-level perturbations (Morris et al., 2020; Zhang et al., 2023) and are commonly used in recent works on adversarial attacks and defense (Dyrmishi et al., 2023; Wang et al., 2023; Bao et al., 2023; Li et al., 2023b;

Findings of the Association for Computational Linguistics: ACL 2026, pages 40954–40971

Text: On the coronavirus I trust the advice of Dr. Fauci over @realDonaldTrump all day long. #COVID19 **Target:** *Anthony Fauci* **Stance:** Favor
Word-level Attack: On the coronavirus I reliable the advice of Dr. Fauci over @realDonaldTrump all day long. #COVID19 **StanceAttack (ours):** In discussions about the coronavirus, Dr. Fauci's insights seem to resonate more convincingly than those of @realDonaldTrump. #COVID19

Table 1: Examples of adversarial attacks on stance detection data (ours is with ChatGPT).

Waghela et al., 2024). Another effective method is the human-generated attack (Jia and Liang, 2017; Nie et al., 2020), where human annotators generate adversarial examples. However, the process of human annotation is time-consuming and labor-intensive, which may not be feasible for all tasks.

In an effort to address the aforementioned problems, in this work, we propose *StanceAttack*, the first adversarial attack method for stance detection aimed at understanding its vulnerabilities. Motivated by the remarkable success of large language models (LLMs) such as ChatGPT reasoning, text understanding and text generation (Wang et al., 2022; Ouyang et al., 2022; Min et al., 2023; Zhou et al., 2023), we develop a Chain-of-Thought prompting technique with ChatGPT to generate adversarial examples through paraphrasing texts from stance detection datasets. We iteratively prompt ChatGPT to produce adversarial examples until successfully deceiving the stance detection models. Our approach provides a more rigorous test for stance detection models, allowing for a deeper understanding of model weaknesses and guiding the development of more robust stance detection models. Table 1 provides examples of adversarial attacks with word replacement and ours with ChatGPT. Our contributions are:

- We propose *StanceAttack*, the first adversarial attack approach for stance detection, and design a novel Chain-of-Thought prompting based on ChatGPT that first performs a semantic analysis between the text and stance target, and then generates human-like adversarial examples aimed at preserving the semantic meaning of the original texts.
- Extensive results on two stance datasets shows that our approach effectively attacks well-trained stance detection models, achieving much higher success rates with significantly fewer queries than baseline methods. Human evaluations confirm that our method produces high-quality adversarial examples that maintain semantic meanings of original texts and read naturally, adhering to grammatical rules.
- We perform adversarial training with a combination of original training set and a small adversarial set and the results demonstrate a significant increase in the robustness of stance detection models.

2 Related Work

Stance Detection Stance detection has garnered considerable attention in recent years (Mohammad et al., 2016; Conforti et al., 2020; Allaway and McKeown, 2020; Glandt et al., 2021). Pretrained language models are extensively used for this task through fine-tuning such as BERT (Allaway and McKeown, 2020; Kaffee et al., 2023), BERTweet (Li et al., 2021, 2023a) and their variants (Liu et al., 2021; Liang et al., 2022; Luo et al., 2022; Hanley and Durumeric, 2023). Additionally, Li et al. (2023c) fine-tune BART for zero-shot stance detection, leveraging pretrained knowledge from natural language inference (NLI) to enhance stance detection. More recently, Large Language Models are increasing explored for stance detection (Weinzierl and Harabagiu, 2024; Zhang et al., 2024; Garg and Caragea, 2024), but their performance is similar with that of BERT-like models or even worse. Recent work has also explored structure-only or structure-dominant approaches for stance detection (Pick et al., 2022; Li et al., 2018). These studies typically focus on scenarios with clean data, assuming environments free from adversarial threats. In practice, however, attackers can create adversarial inputs that easily compromise well-trained stance detection models. Schiller et al. (2021) investigate "blind attacks" for stance detection, generating adversarial examples without utilizing classifier feedback. Unlike previous efforts, our approach pioneers "decision-based attacks" (Moraffah and Liu, 2024), which iteratively use victim model predictions (i.e., models targeted by adversarial attacks) to verify adversarial examples and continue attacking until the model fails.

Adversarial Attacks for NLP Language models, like many machine learning systems, are vulnerable to adversarial attacks (Li et al., 2017; Alzantot et al., 2018; Liang et al., 2018; Zhong et al., 2020; Qiu et al., 2022; Wang et al., 2021; Goyal et al., 2023; Shayegani et al., 2023). Recent studies show that PLMs can be deceived by adversarial examples that, while retaining the semantic integrity of the

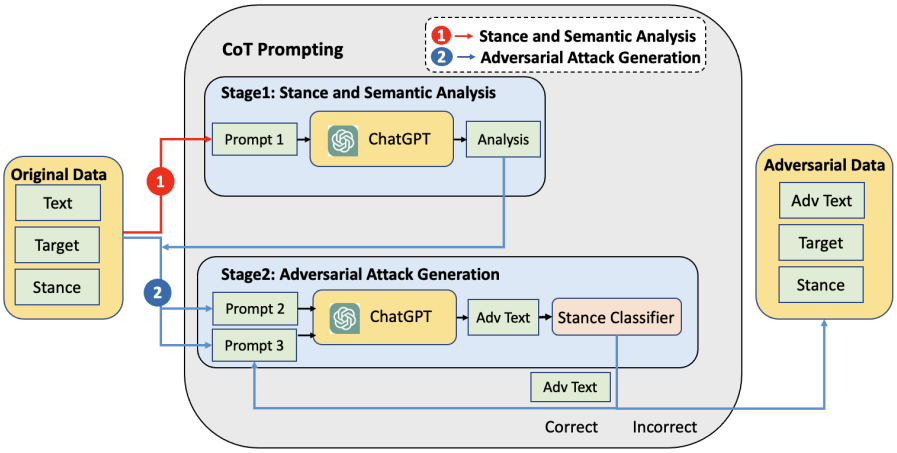

Figure 1: The overall framework of *StanceAttack*.

original text, lead to incorrect outputs (Jin et al., 2020; Li et al., 2020; Wang et al., 2021; Zhang et al., 2023; Nakamura et al., 2023). Adversarial techniques vary primarily in the granularity of text manipulation: character-level, word-level, and sentence-level. Character-level attacks (Ebrahimi et al., 2018; Eger et al., 2019; Chang et al., 2023) manipulate the text by inserting, deleting, flipping, replacing, or swapping characters, often introducing typographical errors that are easily detected due to their unnatural appearance (Dyrmishi et al., 2023). Sentence-level attacks (Ribeiro et al., 2018; Iyyer et al., 2018; Wang et al., 2020; Chen et al., 2021) typically involve inserting new sentences or paraphrasing existing ones using encoder-decoder models (Wang et al., 2020).

Word-level attacks, which involve adding, removing, or modifying words within sentences, stand out as particularly effective (Ren et al., 2019; Garg and Ramakrishnan, 2020; Nguyen et al., 2020). These attacks generally achieve higher success rates than sentence-level attacks (Zeng et al., 2021) and maintain a more natural flow compared to character-level attacks, which can disrupt readability with obvious typos (Ebrahimi et al., 2018; Zhang et al., 2023; Wang et al., 2023). Allaway et al. (2021) study word-level adversarial attacks by swapping words based on sentiment score. Word-level attacks have been widely investigated in recent works on adversarial attacks and defense (Li et al., 2020; Ye et al., 2022; Wang et al., 2023; Bao et al., 2023; Li et al., 2023b; Liu et al., 2024; Waghela et al., 2024). Another significant category is human-generated attacks (Naik et al., 2018; Bartolo et al., 2020; Nie et al., 2020; Kiela et al., 2021; Zhou et al., 2023; Chang et al., 2024), where human annotators are asked to generate the adversarial examples to attack the model. However, human annotation is time-consuming and labor-intensive, making it less practical for large-scale applications (Wang et al., 2021). In our study, we utilize large language models (LLMs) and Chain-of-Thought prompting (Wei et al., 2022) with ChatGPT in an innovative way to generate adversarial attacks that deceive state-of-the-art stance detection models while preserving semantic and grammatical integrity, providing a scalable alternative to human-generated attacks, greatly reducing time and effort.

3 StanceAttack

In this section, we introduce *StanceAttack*, our LLM-based adversarial attack approach for stance detection. Our method consists of two main steps: ❶ **Stance and Semantic Analysis.** We first prompt ChatGPT to perform stance and semantic analysis on the original data. This analysis serves as the foundation for generating adversarial examples. ❷ **Adversarial Attack Generation.** We use insights from the stance and semantic analysis to guide ChatGPT in a Chain-of-Thought approach, prompting it to create adversarial samples designed to attack well-trained stance detection models. The framework of *StanceAttack* is depicted in Figure 1.

3.1 Problem Formulation

Suppose we are given an original stance detection dataset D= $\{(x_i, t_i, y_i)\}_{i=1}^{N}$, where x_i is the text, t_i is the target and $y_i \in$ {Against, Favor, Neutral} is the stance label. We also have a trained stance detection model M whose objective is to predict y_i given x_i and t_i. Under the black-box scenario, the logit output of M is the only supervision we can get. Our objective is to create adversarial examples by paraphrasing x_i into x_i^{adv}, such that x_i^{adv} retains the same semantic meaning as x_i but leads M to

make an incorrect prediction different from y_i.

Algorithm 1 ***StanceAttack***

Stance and semantic analysis
Input: Original data where model correctly predict $D = [(x_0, t_0, y_0), (x_1, t_1, y_1), \ldots]$, victim model M, attack model ChatGPT, max query numbers K.
Output: Adversarial examples $D^{adv} \leftarrow \{\}$
$A \leftarrow \{\}$ //Analysis results
for $x_i, t_i, y_i \in D$ **do**
  $a_i = ChatGPT(P_1(x_i, t_i, y_i))$
  Add a_i to A //Prompt ChatGPT with prompt P1 for stance and semantic analysis.
end for
Adversarial Attack Generation
for $(x_i, t_i, y_i), a_i \in D, A$ **do**
  x_i^{adv}=ChatGPT($P_2(x_i, t_i, y_i, a_i)$) //Prompt with P2 to attack x_i.
  query number Q=0
  while Q<K **do**:
    $pred = M(x_i^{adv}, t_i)$ //Make stance prediction with model M.
    if $pred \neq y_i$ **then**
      Add (x_i^{adv}, t_i, y_i) to D^{adv} //Success attack.
      break
    else
      $\hat{x}_i^{adv}$=x_i^{adv}
      x_i^{adv}=ChatGPT($P_3(x_i, t_i, y_i, \hat{x}_i^{adv})$) //Integrate last unsuccessful attack and prompt with P3 for the next attack.
    end if
    Q+=1
  end while
end for
return D^{adv}

In our StanceAttack framework, we focus specifically on crafting adversarial attacks for the "favor" and "against" categories because in practice, an adversary is more likely to attack instances of "favor" and "against" and turn them into "against" and "favor" instances, respectively (e.g., to change the result on a controversial topic). We also focus on attacking data that the stance detection model M can correctly classify (because on the other data, M already fails to provide a correct prediction).

3.2 Our Chain-of-Thought Prompting

Large language models (LLMs) have shown exceptional ability to learn across a range of tasks (Liang et al., 2023; Dai et al., 2023; Ubani et al., 2023; Zhu et al., 2023). Instead of fine-tuning a pre-trained model, these models can be directly used for new tasks by simply adjusting the prompts used during inference. In our research, we use ChatGPT to create adversarial attacks for stance detection. We develop a two-stage method involving Chain-of-Thought (CoT) prompts that help the LLM to grasp complex nuances of stance and semantics, enabling it to generate effective adversarial attacks.

3.2.1 Stance and Semantic Analysis

In this stage, we focus on extracting the stance and semantic meanings conveyed in the original text and its correlation with the target. For each original data (x_i, t_i, y_i), we use prompt P1 (shown in Appendix B) to guide ChatGPT in performing the following analysis: 1) examine the stance correlation between the text x_i and the target t_i resulting in stance y_i; 2) examine semantic of the x_i; and 3) determine key elements influencing the stance expression. We aim to elucidate how the text's language and structure communicate its stance towards the target, which is helpful information for ChatGPT to understand the data and generate the adversarial texts in the next stage. The collected analysis for each example is denoted as a_i.

3.2.2 Adversarial Attack Generation

In this stage, we prompt ChatGPT to generate adversarial text for each original data (x_i, t_i, y_i). Particularly, we use prompt P2 (detailed in Appendix B) to instruct ChatGPT to use the stance and semantic reasoning provided by a_i as a reference to rephrase x_i, implicitly expressing the stance y_i towards the target t_i. We encourage ChatGPT to employ similar expressions, figurative language, or synonyms. Moreover, the rephrased text must preserve the original semantic meaning and stance toward the target while changing structure and vocabulary. Examples of generated analysis and adversarial examples are shown in Table 2. The adversarial text generation is integrated with victim stance detection models for adversarial attack. Specifically, for each original example (x_i, t_i, y_i), we utilize P2 to guide ChatGPT in paraphrasing x_i into x_i^{adv}. The paraphrased text and target pair, (x_i^{adv}, t_i), is then sent to the victim stance detection model M. If M incorrectly predicts the stance other than y_i, we deem the attack successful and include (x_i^{adv}, t_i, y_i) in our adversarial test set. If M still correctly predicts the stance as y_i, we make $\hat{x}_i^{adv} = x_i^{adv}$ to maintain the last unsuccessful adversarial text and instruct ChatGPT to attempt the paraphrasing again x_i^{adv}. In this subsequent attempt, we refine the CoT prompt for the second round, evolving from P2 to P3, by appending the last unsuccessful adversarial text $\hat{x}_i^{adv}$ to the end. We clearly indicate to ChatGPT that this text is a negative example needing further rephrasing (prompt P3 is shown in Appendix B). If after K attempts—where K is a hyperparameter defining the maximum query number—ChatGPT still fails

Ori Text:	If, 6 months later, you're still wearing a mask.....you might as well wear one the rest of your life.
Target:	Wearing a face mask.
Stance:	Against
Analysis:	The stance taken towards face masks in the provided text is clearly against. The implication that wearing a mask for an extended period, such as six months, is negative and suggests a sarcastic view towards continued mask-wearing. The tone of the text is sarcastic and mocking, implying a dismissive attitude towards the idea of prolonged mask-wearing. The word choice adds a sense of exaggeration and ridicule to emphasize the viewpoint against continued mask usage. The use of the ellipsis at the beginning also signifies a tone of disbelief or exasperation. The pivotal words and phrases in the text that convey the against stance towards face masks include "still wearing a mask", and "you might as well wear one the rest of your life." These phrases suggest a negative sentiment towards the prolonged use of face masks, indicating disapproval or opposition.
Adv Text:	If you're going for the marathon of mask-wearing, might as well make it a lifetime commitment.

Table 2: Examples of semantic and stance analysis, along with adversarial text generated by *StanceAttack*.

Dataset	Source	Train	Val	Test	Avg Len
Covid	Tweet	4533	800	800	27.3
VAST	News	13477	1019	1460	101.1

Table 3: Statistics of stance detection datasets.

to deceive M, we conclude that our method cannot successfully attack this example and discontinue further attempts for this example. Algorithm 1 illustrates this process.

4 Datasets and Baselines

In this section, we introduce datasets, baseline, evaluation metrics, and experimental settings.

4.1 Datasets

Covid (Glandt et al., 2021) consists of tweets related to four targets in the Covid domain: *Anthony Fauci*, *Stay-at-Home Orders*, *Wear a Face Mask*, and *Keeping School Closed*.

VAST (Allaway and McKeown, 2020) includes news comments from The New York Times *Room for Debate* section that contains a large number of targets from multiple domains. We use data for the zero-shot task. Dataset splits for the two datasets are presented in Table 3.

4.2 Baseline Adversarial Attacks

We adopt the following state-of-the-art adversarial attacks that are commonly used as adversarial attack methods and baselines (Dyrmishi et al., 2023; Bao et al., 2023; Li et al., 2023b).

TextFooler (Dyrmishi et al., 2023; Jin et al., 2020) prioritizes words by saliency and selects substitutes from word embeddings to efficiently create adversarial examples.

BERTAttack (Dyrmishi et al., 2023; Li et al., 2020) iteratively masks words in sentences, retrieves substitutes from a Masked Language Model, and greedily selects those that most reduce the victim model's confidence scores.

MAYA (Chen et al., 2021) generates adversarial examples by leveraging paraphrase generation based on small PLMs. In our work, we use the sentence-level attack setting for MAYA.

HQA (Liu et al., 2024) minimizes perturbations by reverting substitutions and optimizing remaining changes with synonyms to enhance semantic similarity while meeting adversarial conditions.

4.3 Evaluation Metrics

To evaluate the quality of the generated adversarial samples, we employ both automatic and human evaluation categories. For automatic metrics, which we calculate directly, we use the **success rate** and **average query number**, following previous studies (Li et al., 2020; Jin et al., 2020; Wang et al., 2021). The success rate is defined as the percentage of samples that successfully fail the stance detection model through adversarial attacks. The average query number measures average attack retries per sample. Additionally, we assess the performance of stance detection models specifically for the "Favor" and "Against" classes. We measure this using per-class F1 scores: $\mathbf{F1_{con}}$ for "Against" and $\mathbf{F1_{pro}}$ for "Favor", along with $\mathbf{F1_{macro}}$, the macro-averaged F1 score across these two classes. We also employ human evaluation metrics, including **semantic similarity** and **grammatical correctness** (Li et al., 2020; Jin et al., 2020; Dyrmishi et al., 2023). Semantic similarity measures the percentage of adversarial samples annotated as semantically similar to the original texts. Grammatical correctness is evaluated based on the average grammatical score from three human annotators, aiming to determine if the adversarial sample appears human-like in its construction.

Dataset	Model	Attack	Ori F1$_{macro}$	Att F1$_{macro}$	Success Rate	Query Number
Covid	BERTweet	*StanceAttack* (ours)	0.811	0.047	97.2%	11.9
		BERT-Attack		0.230	80.8%	53.5
		TextFooler		0.349	51.3%	79.5
		HQA		0.339	53.1%	70.2
		MAYA		0.305	62.8%	70.6
	BART	*StanceAttack* (ours)	0.743	0.017	99.2%	8.5
		BERT-Attack		0.284	94.1%	38.0
		TextFooler		0.448	75.3%	67.0
		HQA		0.470	72.2%	76.2
		MAYA		0.462	72.5%	66.7
VAST	BERT	*StanceAttack* (ours)	0.636	0.172	87.5%	29.7
		BERT-Attack		0.513	41.3%	85.1
		TextFooler		0.536	31.7%	92.8
		HQA		0.531	33.0%	69.2
		MAYA		0.545	27.6%	98.0
	BART	*StanceAttack* (ours)	0.705	0.206	76.6%	40.2
		BERT-Attack		0.422	55.9%	79.9
		TextFooler		0.504	51.7%	88.4
		HQA		0.682	42.0%	88.0
		MAYA		0.621	45.8%	98.1

Table 4: Comparison of our *StanceAttack* approach with previous state-of-the-art adversarial attack baselines. Lower Att F1$_{macro}$ Scores Indicate Better Performance.

4.4 Experimental Settings

Victim Models For each dataset, we evaluate two small PLMs that are renowned for their state-of-the-art performance and use them as victim models (i.e., stance prediction models targeted by our adversarial attacks—the M model from §3.1). For the Covid dataset, we utilize **BERTweet** (Nguyen et al., 2020), a RoBERTa-based model fine-tuned for stance prediction (Glandt et al., 2021; Li et al., 2021), and **BART** (Li et al., 2023c), which is initially pretrained on the MNLI dataset (Williams et al., 2018) and fine-tuned for stance detection (Li et al., 2023c). For the VAST dataset, we select **BART** and **BERT** (Devlin et al., 2019), as both models are commonly used for this dataset (Allaway and McKeown, 2020). We select small PLMs as victim models due to their better performance on stance prediction than LLMs (Appendix A).

Training Settings For each dataset, we first fine-tune the stance detection models on the original training set. We then generate adversarial attacks on the original test set (i.e., data labeled with "favor" and "against"), excluding original data that already fails stance detection models. Our adversarial test consists of: 1) original data that fails the model, 2) adversarial data that successfully compromise the model, and 3) adversarial data that fail to compromise the model after K retries, thus, maintaining the same size as the original test set. We compare ChatGPT with LLaMA 3 and Mistral for attack generation, as detailed in Appendix C. We show hyperparameters in Appendix E.

5 Results

In this section, we first compare *StanceAttack* against baseline adversarial methods (5.1). Next, we conduct ablation studies to assess the effectiveness of the CoT components (5.2). We then conduct adversarial training based on *StanceAttack* (5.3) (leveraging a small amount of train and dev examples). Finally, we perform human evaluation to compare the quality of adversarial data (5.4).

5.1 Comparison with Adversarial Baselines

We first evaluate models trained on original training data using test data with adversarial attacks generated by *StanceAttack* and contrast them with baseline methods. Particularly, we allow maximum retries K=100 and compare our approach with baseline methods on the following metrics: 1) F1$_{macro}$ on original test set (denoted as Ori F1$_{macro}$), 2) F1$_{macro}$ on the attacked test set (denoted as Att F1-macro), 3) success rate, and 4) average number of queries. Next, we compare the success rate of *StanceAttack* with baselines as we use different max-query number K.

Results are displayed in Table 4. First, we observe that *StanceAttack* is an effective adversarial attack which consistently achieves a higher success rate across all tested datasets and models over baseline methods. For instance, when attacking BERTweet on the Covid dataset, *StanceAttack* achieves a success rate of 97.2%, which surpasses BERT-Attack by 16.4%. This is further evidenced by the "Att F1$_{macro}$" scores, where *StanceAttack* severely degrades the performance of well-trained

Method	Ori F1$_{macro}$	Att F1$_{macro}$	Success Rate	Query Number
StanceAttack	0.811	0.047	97.20%	11.9
w/o P1		0.239	78.00%	17.6
w/o P3		0.205	81.40%	15.5
CoT 1-stage		0.121	89.80%	14.1

Table 5: Ablation study of our *StanceAttack* approach for BERTweet on the Covid dataset.

Dataset	Model	Train	Original Test			Adversarial Test		
			Con	Pro	All	Con	Pro	All
Covid	BERTweet	Ori	0.805	0.817	0.811	0.024	0.070	0.047
		Ori+Adv	0.766	0.807	0.787	0.643	0.677	0.660
	BART	Ori	0.743	0.742	0.743	0.006	0.029	0.017
		Ori+Adv	0.756	0.772	0.764	0.649	0.662	0.656
VAST	BERT	Ori	0.669	0.603	0.636	0.185	0.158	0.172
		Ori+Adv	0.659	0.611	0.635	0.541	0.438	0.489
	BART	Ori	0.706	0.704	0.705	0.109	0.304	0.206
		Ori+Adv	0.713	0.692	0.702	0.622	0.583	0.603

Table 6: Comparison of model performance on original test set (Original Test) and adversarial test set generated by *StanceAttack* (Adversarial Test) for models trained with original training data (Ori) versus models trained with a combination of original and adversarial training data (Ori+Adv).

models to a much greater extent than the baseline methods do, indicating its effectiveness as a stronger adversarial attack. Secondly, *StanceAttack* is also an efficient adversarial attack method, requiring far fewer queries compared to baselines. For instance, when attacking BART on the Covid dataset, *StanceAttack* only needs fewer than 9 queries to succeed. In contrast, baseline require significantly more queries. Third, the VAST dataset, which consists of text with an average length of over 100 (as shown in Table 3), presents greater challenges for adversarial attacks compared to the shorter Twitter-based Covid dataset with the average length around 27. The longer texts contain more contextual information, making it tougher for adversarial attacks to successfully compromise well-trained models. Despite these challenges, *StanceAttack* still demonstrates robust performance, achieving success rates of 87.5% and 76.6% when attacking BERT and BART, respectively, requiring only 29.7 and 40.2 average queries, significantly outperforming the baseline models. In Appendix F, we provide an analysis examining the causes of stance flips on our adversarial data. Moreover, we investigate impact of K in Appendix G. We compare our approach with training on one dataset and evaluating on another in Appendix H. We also assess model robustness by analyzing prediction changes after our attack in Appendix I. Results on P-STANCE dataset is in Appendix J. We use LLM (Gemini) as victim models for comparison in Appendix K.

5.2 Ablation Study

In this section, we carry out an ablation study to evaluate the effectiveness of different components of CoT prompting including: 1) *StanceAttack* without using prompt P1 for stance analysis, directly prompting ChatGPT to generate attacks (w/o P1); 2) *StanceAttack* with the adversarial attack framework, where ChatGPT uses only the original P2 prompt without incorporating feedback from previous unsuccessful attempts (w/o P3—always using P2); and 3) *StanceAttack* with using a single-stage CoT process that combines semantic analysis and adversarial attack generation into a single prompt (CoT 1-stage—we show the prompt in Appendix B). Experiments are performed using our best-performing BERTweet on the Covid dataset.

Results are shown in Table 5. We find that omitting stance and semantic analysis (w/o P1) significantly reduces the success rate and increases the number of queries needed, suggesting that information from CoT reasoning aids in more efficient generation of adversarial attacks. Additionally, failing to integrate feedback from the last unsuccessful attack into the subsequent prompt (w/o P3) also results in a lower success rate and more queries, indicating that using insights from previous failures can accelerate the identification of successful attacks. Last, the CoT 1-stage approach also demonstrates a lower success rate and a higher query count compared to *StanceAttack* with all P1, P2, and P3, suggesting that dividing the CoT prompt into steps allows the LLM to better comprehend the task and generate more appropriate texts.

5.3 Adversarial Training Results

Adversarial training (Goodfellow et al., 2015), designed to improve the robustness of machine learning models, involves enriching the training and vali-

Dataset	Method	Semantic (%)	Grammar
Covid	*StanceAttack*	83.00	4.41
	BERT-Attack	78.42	3.27
	TextFooler	82.35	3.22
	HQA	82.00	3.22
	MAYA	72.05	2.36
VAST	*StanceAttack*	89.00	4.19
	BERT-Attack	93.94	2.64
	TextFooler	92.16	2.90
	HQA	92.06	3.15
	MAYA	84.10	2.02

Table 7: Evaluation of semantic and grammar metrics for different attack methods.

dation sets with adversarial examples. In our experiments, we generated 700 adversarial examples for each stance detection model's training set and an additional 100 for the validation set using *StanceAttack* (Statistics in Appendix D). These examples were then integrated into the respective original datasets. We then retrained the target models with these augmented adversarial data. We evaluated the robustness of these models by analyzing their F1 scores on both the original and adversarial test sets, as detailed in Table 6. Our results demonstrate notable improvements on the adversarial test sets without compromising their performance on the original test sets. For instance, BART recorded a 39.7% increase in $F1_{macro}$ on the VAST adversarial test set, while maintaining its original test set performance.

5.4 Human Evaluation

To further assess the quality of adversarial examples generated by our *StanceAttack* method, we conducted human evaluations, following the approaches of previous works (Li et al., 2020; Dyrmishi et al., 2023; Jin et al., 2020). We organized the evaluation around two primary criteria: semantic similarity to the original text and grammatical correctness. For the semantic similarity assessment, annotators were given pairs of original and adversarial texts. These adversarial samples, sourced from *StanceAttack* and baseline methods, were presented in random order. Annotators evaluated whether the original and adversarial texts retained similar meanings. In terms of grammatical correctness, human annotators rated the adversarial texts on a 1 to 5 scale, following the guidelines outlined in (Jin et al., 2020; Li et al., 2020). We evaluated 100 original and adversarial texts from each attack method using both the Covid and VAST datasets. Each sample was reviewed by three annotators. For semantic similarity, we determined the outcome based on the majority opinion of the annotators, and for grammar assessment, we calculated the average score they provided. Human evaluation results confirm that the paraphrased texts are stance-preserving (see Appendix L). Annotation platform and instructions are in Appendix M.

Results shown in Table 7 reveal that adversarial samples created by *StanceAttack* consistently score higher on grammatical correctness across both datasets compared to those from the two baseline methods. This indicates that *StanceAttack* is capable of generating more human-like adversarial samples with proper grammar. Additionally, for the Covid dataset, our method exhibits a higher semantic similarity ratio than the baseline models. However, for the VAST dataset, the ratio is slightly lower than the word-level attacking baselines (i.e., BERT-Attack, TextFooler, and HQA.). This is because the longer texts in the VAST dataset pose extra challenge for our rephrasing-based attacking method. The extensive contextual information in longer texts makes it more difficult to maintain semantic similarity when attempting to fail the model. In contrast, word-level attacks typically involve word replacements, and tend to retain semantic meaning more easily. Despite these challenges, StanceAttack still achieves only marginally lower performance, highlighting its overall effectiveness relative to baseline methods. Notably, our method still shows much better performance when comparing with the sentence-level attack MAYA.

6 Conclusion

In this work, we introduce StanceAttack, the first adversarial attack method for stance detection that utilizes ChatGPT's advanced text generation capabilities. Our method designs a Chain-of-Thought prompting strategy that first performs a semantic analysis of the original text and stance target and then creates adversarial examples that deceive robust stance detection models while preserving the original text's semantic meaning and grammatical integrity. Experimental results across two datasets show that our approach outperforms conventional adversarial attack methods in success rate and efficiency. Through adversarial training, we enhance model robustness against such attacks. This study not only highlights vulnerabilities in stance detection models but also provides a comprehensive framework for enhancing their security, thereby contributing to the development of more robust stance detection models. We make our code and adversarial data publicly available.

Acknowledgements

We thank the US National Science Foundation for funding from the grant IIS-2107487 which supported the research and the computation in this study.

Limitations

Our focus is specifically on crafting adversarial attacks for the "favor" and "against" categories because, in practice, an adversary is more likely to attack instances of "favor" and "against" and turn them into "against" and "favor" instances, respectively (e.g., to change the result on a controversial topic), whereas manipulating "neutral" instances is generally less meaningful and of limited strategic interest to an adversary. Thus, excluding the "neutral" class might be seen as a limitation. However, the neutral class and its generation for stance detection deserves a special focus since LLMs still struggle at generating samples for this class and, if used, it may remain unclear whether the models fail on those samples because of their adversarial nature or because the overall generation of this class is not satisfactory. We plan to investigate this thoroughly in the future.

Ethical Statement

Our study adheres to ethical guidelines by generating adversarial examples through paraphrasing existing texts from publicly available benchmark datasets without introducing any additional personally identifiable information. This ensures compliance with data protection standards and minimizes privacy risks. The alterations are designed solely to test model vulnerabilities. We emphasize the use of this research to improve system security and advocate for responsible application of adversarial techniques.

References

Abeer ALDayel and Walid Magdy. 2021. Stance detection on social media: State of the art and trends. *Inf. Process. Manage.*, 58(4).

Emily Allaway and Kathleen McKeown. 2020. Zero-Shot Stance Detection: A Dataset and Model using Generalized Topic Representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8913–8931, Online. Association for Computational Linguistics.

Emily Allaway, Malavika Srikanth, and Kathleen McKeown. 2021. Adversarial learning for zero-shot stance detection on social media. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4756–4767, Online. Association for Computational Linguistics.

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.

Rong Bao, Rui Zheng, Liang Ding, Qi Zhang, and Dacheng Tao. 2023. CASN:class-aware score network for textual adversarial detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 671–687, Toronto, Canada. Association for Computational Linguistics.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Guoqin Chang, Haichang Gao, Zhou Yao, and Haoquan Xiong. 2023. Textguise: Adaptive adversarial example attacks on text classification model. *Neurocomputing*, 529:190–203.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

Yangyi Chen, Jin Su, and Wei Wei. 2021. Multi-granularity textual adversarial attack with behavior cloning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4511–4526, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wang Chunling, Zhang Yijia, Yu Xingyu, Liu Guantong, Chen Fei, and Lin Hongfei. 2023. Adversarial network with external knowledge for zero-shot stance detection. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics*, pages 824–835, Harbin, China. Chinese Information Processing Society of China.

Costanza Conforti, Jakob Berndt, Marco Basaldella, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. 2021. Adversarial training for news stance detection: Leveraging signals from a multi-genre corpus. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 1–7, Online. Association for Computational Linguistics.

Costanza Conforti, Jakob Berndt, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. 2020. Will-they-won't-they: A very large dataset for stance detection on Twitter. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1724, Online. Association for Computational Linguistics.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

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.

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

Krishna Garg and Cornelia Caragea. 2024. Stanceformer: Target-aware transformer for stance detection. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4969–4984, Miami, Florida, USA. 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.

Kyle Glandt, Sarthak Khanal, Yingjie Li, Doina Caragea, and Cornelia Caragea. 2021. Stance detection in COVID-19 tweets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1596–1611, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. 2023. A survey of adversarial defenses and robustness in nlp. *ACM Comput. Surv.*, 55(14s).

Hans Hanley and Zakir Durumeric. 2023. TATA: Stance detection via topic-agnostic and topic-aware embeddings. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11280–11294, Singapore. 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 and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. 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.

Lucie-Aimée Kaffee, Arnav Arora, and Isabelle Augenstein. 2023. Why should this article be deleted? transparent stance detection in multilingual Wikipedia editor discussions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5891–5909, Singapore. Association for Computational Linguistics.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Dilek Küçük and Fazli Can. 2020. Stance detection: A survey. *ACM Comput. Surv.*, 53(1).

Ang Li, Bin Liang, Jingqian Zhao, Bowen Zhang, Min Yang, and Ruifeng Xu. 2023a. Stance detection on social media with background knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15703–15717, Singapore. Association for Computational Linguistics.

Chang Li, Aldo Porco, and Dan Goldwasser. 2018. Structured representation learning for online debate stance prediction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3728–3739, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Linyang Li, Demin Song, and Xipeng Qiu. 2023b. Text adversarial purification as defense against adversarial attacks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 338–350.

Yingjie Li, Tiberiu Sosea, Aditya Sawant, Ajith Jayaraman Nair, Diana Inkpen, and Cornelia Caragea. 2021. P-stance: A large dataset for stance detection in political domain. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2355–2365, Online. Association for Computational Linguistics.

Yingjie Li, Chenye Zhao, and Cornelia Caragea. 2023c. Tts: A target-based teacher-student framework for zero-shot stance detection. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 1500–1509, New York, NY, USA. Association for Computing Machinery.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4208–4215.

Bin Liang, Qinglin Zhu, Xiang Li, Min Yang, Lin Gui, Yulan He, and Ruifeng Xu. 2022. JointCL: A joint contrastive learning framework for zero-shot stance detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Yuanyuan Liang, Jianing Wang, Hanlun Zhu, Lei Wang, Weining Qian, and Yunshi Lan. 2023. Prompting large language models with chain-of-thought for few-shot knowledge base question generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4329–4343, Singapore. Association for Computational Linguistics.

Han Liu, Zhi Xu, Xiaotong Zhang, Feng Zhang, Fenglong Ma, Hongyang Chen, Hong Yu, and Xianchao Zhang. 2024. Hqa-attack: Toward high quality black-box hard-label adversarial attack on text. *Advances in Neural Information Processing Systems*, 36.

Rui Liu, Zheng Lin, Yutong Tan, and Weiping Wang. 2021. Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yun Luo, Zihan Liu, Yuefeng Shi, Stan Z. Li, and Yue Zhang. 2022. Exploiting sentiment and common sense for zero-shot stance detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7112–7123, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2):1–40.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Raha Moraffah and Huan Liu. 2024. Exploiting class probabilities for black-box sentence-level attacks. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1557–1568, St. Julian's, Malta. Association for Computational Linguistics.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Mutsumi Nakamura, Santosh Mashetty, Mihir Parmar, Neeraj Varshney, and Chitta Baral. 2023. LogicAttack: Adversarial attacks for evaluating logical consistency of natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13322–13334, Singapore. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Ron Korenblum Pick, Vladyslav Kozhukhov, Dan Vilenchik, and Oren Tsur. 2022. Stem: unsupervised structural embedding for stance detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11174–11182.

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

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.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. Stance detection benchmark: How robust is your stance detection? *KI-Künstliche Intelligenz*, pages 1–13.

Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. 2023. Survey of vulnerabilities in large language models revealed by adversarial attacks. *arXiv preprint arXiv:2310.10844*.

Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. Zeroshotdataaug: Generating and augmenting training data with chatgpt. *arXiv preprint arXiv:2304.14334*.

Hetvi Waghela, Sneha Rakshit, and Jaydip Sen. 2024. A modified word saliency-based adversarial attack on text classification models. *arXiv preprint arXiv:2403.11297*.

Boxin Wang, Hengzhi Pei, Boyuan Pan, Qian Chen, Shuohang Wang, and Bo Li. 2020. T3: Tree-autoencoder constrained adversarial text generation for targeted attack. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6134–6150, Online. Association for Computational Linguistics.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Wenxuan Wang, Wenxiang Jiao, Yongchang Hao, Xing Wang, Shuming Shi, Zhaopeng Tu, and Michael Lyu. 2022. Understanding and improving sequence-to-sequence pretraining for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2591–2600, Dublin, Ireland. 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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Maxwell Weinzierl and Sanda Harabagiu. 2024. Tree-of-counterfactual prompting for zero-shot stance detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–880, Bangkok, Thailand. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Muchao Ye, Jinghui Chen, Chenglin Miao, Ting Wang, and Fenglong Ma. 2022. Leapattack: Hard-label adversarial attack on text via gradient-based optimization. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2307–2315.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. OpenAttack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 363–371, Online. Association for Computational Linguistics.

Jianping Zhang, Yung-Chieh Huang, Weibin Wu, and Michael R Lyu. 2023. Towards semantics-and domain-aware adversarial attacks. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 536–544.

Zhao Zhang, Yiming Li, Jin Zhang, and Hui Xu. 2024. LLM-driven knowledge injection advances zero-shot and cross-target stance detection. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 371–378, Mexico City, Mexico. Association for Computational Linguistics.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural deepfake detection with factual structure of text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2461–2470, Online. Association for Computational Linguistics.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*.

Yiming Zhu, Peixian Zhang, Ehsan-Ul Haq, Pan Hui, and Gareth Tyson. 2023. Can chatgpt reproduce human-generated labels? a study of social computing tasks. *arXiv preprint arXiv:2304.10145*.

A Compare PLMs with LLMs on Stance Prediction

We compare fine-tuned PLMs that we selected with prompted LLMs on VAST and Covid dataset. The results are shown in Table 9. We can observe that in most cases PLMs show much better performance than LLMs. Hense we mainly foccus on PLMs as victim models due to their overall better performance on stance prediction.

B Prompt for ChatGPT

Prompts that we used for *StanceAttack* are shown in Table 8. At the first stage, we use prompt1 for ChatGPT to generate stance and semantic correlation. For the second stage, we use the prompt 2 to ask ChatGPT to generate adversarial text. We use prompt 3 to introduce previously unsuccessful adversarial text into the prompt for the next attempt. For P1, we focus on generating analysis. We first explain the task and the specific aspects we want the model to analyze, such as stance correlation and semantics. We then use ChatGPT to generate the prompt based on these requirements. For P2, we specify the requirement, such as rephrasing the text while maintaining semantic meanings or utilizing the provided analysis, and then prompt ChatGPT to generate the corresponding prompt. P3 follows the same approach as P2 but includes an additional sentence at the end, incorporating feedback from the previous attempt into the next iteration. This process is detailed in Section 3.2 and Table 8.

C ChatGPT vs. other LLMs for Attack Generation

Our initial manual investigation showed that ChatGPT generates higher quality texts compared to LLaMA3 and Mistral. We include an example and the generated text for each model in Table 10, where it is evident that ChatGPT produces text with

Prompt1:	In this task, we will delve into understanding the stance and the semantic meanings conveyed in the provided text, focusing on its correlation with the specified target. By breaking down the analysis into clear, logical steps, we aim to elucidate how the text's language and structure communicate its stance towards the target. This detailed examination will not venture into rephrasing but will instead concentrate on dissecting and interpreting the original content. 1.Analyze the Stance Towards the Target: Identify and analyze the stance the text takes towards the specified target. Is the stance favor, against, or neutral? Determine the nature of the sentiment being expressed towards the target. 2.Semantic Analysis of the Text: Analyze the semantic layers within the text. What words or phrases are pivotal in conveying the stance? How do these elements interplay to express favor, against, or neutral towards the target? 3.Key Elements Influencing Stance Expression: Identify key elements and linguistic features in the text that influence how the stance is expressed. Consider tone, word choice, and any figurative language used. How do these contribute to the overall stance orientation? 4.Correlation Between Text and Stance: Examine how the identified semantic elements and linguistic features correlate with the expressed stance. How effectively does the text communicate its position regarding the target? Are there subtle cues or overt statements that make this stance evident? Please apply this analytical process to the following text, target and the stance correlation, focusing on the semantic meaning and the stance correlation: Original Text: [TEXT] Target: TARGET Stance: STANCE
Prompt2:	Rephrase the following text to subtly express the same stance towards the specified target. The rephrasing should aim to convey the message in a manner that is not straightforward for stance detection algorithms to classify. Consider using sarcasm, figurative language, synonyms, similar expressions, indirect language, metaphors, or summarizing the sentiment in a nuanced way that retains the original stance towards the target but changes the sentence structure and vocabulary significantly. Please ensure the rephrased text remains coherent and can be understood by a human reader as maintaining the original stance towards the given text without directly mimicking the original text's style or terminology. Before rephrasing, a detailed analysis based on the semantic of the original text and the stance correlation between the text and the target will be provided. This analysis should guide the rephrasing process, ensuring that the nuanced stance and semantic meaning is effectively communicated through alternative expressions and structures. Original Text: [TEXT] Target: [TARGET] Stance: [STANCE] Semantic Analysis and Stance Correlation: [ANALYSIS] Remember, the rephrased version should be creative and indirect enough to potentially obscure the stance from automated classifiers while remaining clear to a human audience.", where [TEXT], [TARGET], [STANCE], and [ANALYSIS] represent text, target, stance labels, and the analysis generated in the first stage, respectively.
Prompt3:	Rephrase the following text to subtly express the same stance towards the specified target. The rephrasing should aim to convey the message in a manner that is not straightforward for stance detection algorithms to classify. Consider using sarcasm, figurative language, synonyms, similar expressions, indirect language, metaphors, or summarizing the sentiment in a nuanced way that retains the original stance towards the target but changes the sentence structure and vocabulary significantly. Please ensure the rephrased text remains coherent and can be understood by a human reader as maintaining the original stance towards the given text without directly mimicking the original text's style or terminology. Before rephrasing, a detailed analysis based on the semantic of the original text and the stance correlation between the text and the target will be provided. This analysis should guide the rephrasing process, ensuring that the nuanced stance and semantic meaning is effectively communicated through alternative expressions and structures. Original Text: [TEXT] Target: [TARGET] Stance: [STANCE] Semantic Analysis and Stance Correlation: [ANALYSIS] Remember, the rephrased version should be creative and indirect enough to potentially obscure the stance from automated classifiers while remaining clear to a human audience.", where [TEXT], [TARGET], [STANCE], and [ANALYSIS] represent text, target, stance labels, and the analysis generated in the first stage, respectively. Here is a bad example, where the rephrased text is too similar to the original text: "[ADV TEXT PREV]"
Prompt for CoT 1-step:	In order to subtly rephrase the following text to express the same stance towards the specified target, let's first analyze the input example in a step-by-step manner. This will involve identifying the stance correlation between the original text and the target, analyzing the semantic meaning of the text, and then creatively rephrasing the text. This process aims to make the message less straightforward for stance detection algorithms to classify, while still clear to a human audience. 1.Identify the Stance Correlation: Examine how the original text's stance (favor, against, neutral) aligns with the specified target. Consider the explicit or implicit cues in the text that reveal this stance. 2.Analyze the Semantic Meaning: Dive deeper into the text to understand its semantic meaning. Look for synonyms, similar expressions, and the overall sentiment that supports the identified stance towards the target. 3.Creative Rephrasing the original text: Based on the understanding of the stance and semantic meaning, think of ways to rephrase the text. Use using figurative language, sarcasm, synonyms, similar expressions, indirect language, metaphors, or summarize the sentiment in a nuanced way that retains the original stance but significantly changes the sentence structure and vocabulary. Ensure the rephrased text is coherent and can be understood by a human reader as maintaining the original stance towards the given text, without directly mimicking the original text's style or terminology. Let's start with this process: Original Text: [TEXT] Target: [TARGET] Stance: [STANCE] Remember, the goal of the rephrased version is to be creative and indirect enough to potentially obscure the stance from automated classifiers while remaining clear to a human audience. Remember, the answers should strictly follow the format of: 1): ANSWER to Identify the Stance Correlation Between the Text and the Target; 2): Analyze the Semantic Meaning of the text; 3): ANSWER to Creative Rephrasing of the text.

Table 8: Prompts that we used in *StanceAttack*. TEXT, TARGET, STANCE, and ANALYSIS represent text, target, stance, and stance and semantic analysis, respectively.

	Model	Con	Pro	All
Covid	BART	0.743	0.742	0.743
	BERTweet	0.805	0.817	0.811
	Gemini	0.636	0.672	0.654
	LLaMA 3	0.569	0.531	0.550
	Mistral	0.592	0.476	0.534
VAST	BERT	0.669	0.603	0.636
	BART	0.706	0.704	0.705
	Gemini	0.635	0.661	0.648
	LLaMA 3	0.595	0.582	0.588
	Mistral	0.537	0.457	0.497

Table 9: Compare PLMs with LLMs on Stance Prediction.

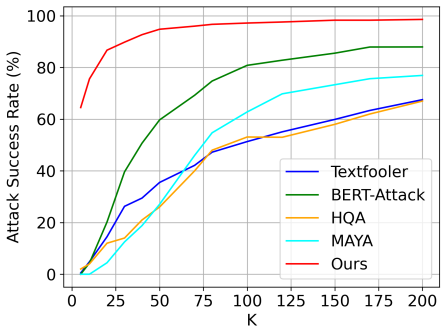

Figure 2: Comparison of our *StanceAttack* approach with baseline attack methods with different K. Attacks are performed based on BERTweet for Covid dataset.

better quality and preserved semantic meanings. We chose ChatGPT due to its superior text generation capabilities. To mimic a realistic scenario, when we have an attacker, the attacker would prefer to use a stronger model that has a better chance of successfully attacking a stance detection model. Our aim is to evaluate a model that has a high likelihood of effectively challenging a stance model.

D Statistics of Original and Adversarial Test Sets

We show the detailed statistics of the adversarial test set and the original test set in Table 11.

E Hyperparameters

For our target stance detection models, we utilize the BART large model[1], pretrained on the MNLI dataset, and follow established protocols ((Li et al., 2023c)) to fine-tune the BART encoder using a stance detection dataset. Due to memory constraints, the BART decoder is excluded. Similarly, we employ both the BERTweet-large[2] and BERT-base[3], following prior research (Glandt et al., 2021; Li et al., 2021). We follow hyperparameters used in previous works to train stance detection models. The learning rate of is set to 1e-5. AdamW (Loshchilov and Hutter, 2019) is utilized as the optimizer. The mini-batch is set to 64. The model is trained for 20 epochs and 4 epochs for Covid and VAST dataset, respectively, with early stopping and the patience is 5. We use gpt-3.5-turbo version of ChatGPT for CoT prompting. The versions of Llama, Mistral, and Gemini are as follows: Llama-3 8B, Mistral 7B, and Gemini 2.0 Pro.

[1] https://huggingface.co/facebook/bart-large-mnli

[2] https://huggingface.co/vinai/bertweet-large

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

F Examples of stance flips

In Table 12, we show examples where our attack successfully causes the stance flips for PLMs. In the first example, the rephrased text preserves the original against stance by expressing concern over the impact of school closures on vulnerable children but uses softer, more nuanced language. The model (BERTweet) misclassifies it as favor due to its reliance on surface-level cues like political alignment and tentative phrasing, which obscure the underlying stance. This highlights the model's weakness in handling implicit or indirect expressions of opinion. In the second example, the rephrased text retains the original argument against mandatory voting by emphasizing concerns over uninformed voters, manipulation through commercial influence, and the need for campaign finance reform. However, the inclusion of the word "fascinating" in the first sentence introduces a positive tone that likely confuses the model (BART), leading to an incorrect "favor" prediction. However, the overall text clearly maintains an "against" stance toward mandatory voting by emphasizing discernment in abstaining and criticizing mandatory participation. This example reveals that the model tends to focus on superficial cues like sentiment-laden words rather than fully capturing the underlying semantic meaning, highlighting a key vulnerability in PLMs' stance detection.

G Comparison with Baselines on Different K

We also compare *StanceAttack* with baseline methods by analyzing the attack success rate at various maximum query limits (K). Our goal is to determine how effectively our approach can compromise stance detection models at different K

Original Text	@ChandlerTV I will never understand the anti-mask movement.
Rephrased Text (ChatGPT)	Unveiling the mysteries of the anti-mask movement seems like an everlasting riddle to me. #BafflingBeliefs.
Rephrased Text (LLaMA3)	I must admit, I find it intriguing that some individuals might be hesitant towards face masks.
Rephrased Text (Mistral)	The history and evolution of anti-mask wearing practices are fascinating to explore.

Table 10: Examples of LLaMA 2 and LLaMA 3 stance prediction when using 10 in-context examples per class.

Dataset	Model	Adv				Ori
		AF	SA	FA	All	All
Covid	BERTweet	84	410	12	506	506
	BART	134	369	3	506	506
VAST	BERT	344	505	92	941	941
	BART	277	494	170	941	941

Table 11: Statistics of adversarial (Adv) and original (Ori) test set. "AF" represents test data that can already fail the stance detection model. "SA" represents adversarial test data that successfully compromise the model. "FA" represents adversarial test data that failed to compromise the model. "All" represents all test data.

values. We carried out these experiments using the best-performing BERTweet model on the Covid dataset, with the results depicted in Figure 2. Our approach consistently outperforms the baselines as K increases. For example, with a maximum of only 5 retries, *StanceAttack* achieves a 62% success rate, substantially higher than baselines, which show negligible impact on the models. At the upper limit of 200 queries, our success rate soars to 98.9%, markedly surpassing baseline methods. These results highlight the effectiveness of our rephrasing-based adversarial attacks, which leverage large language models (LLMs), in comparison to traditional word-level attack strategies in stance detection tasks.

H Comparison with Out-of-topic Setting

We conducted out-of-topic cross-domain experiments to further evaluate the robustness of the stance detection model. Specifically, we trained the stance detection model on one dataset and tested it on a different dataset (e.g., training BERTweet on Covid and testing it on VAST), and vice versa. This approach allowed us to assess the model's performance in out-of-topic scenarios and compare it with the results from our adversarial attacks.

The results are summarized in the Table 13, where 'In-domain Ori' represents the evaluation results on the original test set when the model is trained on the original training set, and 'In-domain Att (attacked)' represents the evaluation results on the adversarial test set when the model is trained on the original training set for the in-domain setting. 'Cross-domain Ori' represents the performance of the model on the original test set when trained on the training set of the other dataset. We observed that the model's performance on the adversarial test data (In domain Att) was significantly lower than in the cross-domain setting across all configurations. This indicates that adversarial attacks effectively expose vulnerabilities in the model, as the text modifications introduced by the attacks create more challenging scenarios than those arising naturally in cross-domain data. These findings highlight the inherent limitations of the train-test probability alignment under adversarial conditions and further underscore the value of our proposed method.

I Robustness Analysis of Model Predictions

In our experiments, the models are trained in a 3-class setting, allowing them to predict "neutral" as a label, even though there are no gold neutral instances in the test set. We analyzed the success cases of fooling the BART model on the VAST dataset and categorized them into two types: predictions that changed to "neutral" and those that resulted in a direct stance flip between "favor" and "against". Specifically, 29.3% of the successful adversarial examples caused the model to predict "neutral" instead of the original stance, while 70.7% resulted in a direct stance flip. We calculate the percentage of fooling based on the total number of adversarial examples that led to any change in the model's output, regardless of whether the change was to "neutral" or a direct stance flip. This breakdown provides a clearer understanding of how the adversarial examples affect the model's predictions.

J Results on P-STANCE dataset

We also conduct experiments on the P-STANCE dataset, and the results are presented in Table 14.

Ori Text:	@ZacBissonnette That video, my god. I'm as progressive as the next person and I dearly hope Trump will lose, but I can't remember the last time I watched such a cynical, fear-mongering piece of propaganda. Keeping schools closed will be devastating for our most vulnerable children.
Target:	school closures
Stance:	Against
Adv Text:	After witnessing that video, I must say, it certainly left an impression. While I generally lean towards the forward-thinking side of things and cross my fingers for Trump's downfall, I can't help but feel unsettled by the overly pessimistic, fear-inducing nature of that piece. The decision to prolong school closures may pose significant challenges for our most at-risk youngsters.
Prediction:	Favor
Ori Text:	Mandatory voting would make matters worse. Most of those not voting are making a wise decision since most are not well informed on the issues. Forcing them to vote would mean that commercial advertisements would influence elections even more than they do now. These media buys are financed by special interest groups whose interests generally do not serve the nation as a whole. The money they spread corrupts the system and is at the root of our current political mess. We need to start with real campaign finance reform that not only restricts how much can be given but also how much can be spent. It requires a constitutional amendment so it will be difficult. It can still happen if the public becomes aware of how bad the situation is and that a real solution is available. This is not a left or right issue. It is one that goes to core values of all Americans.
Target:	mandatory voting
Stance:	Against
Adv Text:	It's fascinating how some believe mandatory voting would be a panacea when, in reality, those abstaining are exercising discernment. Enforcing their participation would only amplify the cacophony of commercial persuasions in our electoral process. These monetary tugs-of-war, orchestrated by vested interests, poison the very roots of our political landscape. Genuine change must begin with comprehensive campaign finance reforms that not only cap contributions but also rein in expenditures. Achieving this will be no walk in the park, requiring a constitutional amendment and widespread awareness of the urgent need for reform. This isn't about partisan divides; it's about upholding the fundamental values that unite all Americans.
Prediction:	Favor

Table 12: Examples of stance flips in PLMs under adversarial attack.

Dataset	Model	In domain Ori	In domain Att	Cross-domain Ori
Covid	BERTWeet	.811	.047	.345
	BART	.743	.017	.465
VAST	BERT	.636	.172	.473
	BART	.705	.206	.535

Table 13: Comparison of our *StanceAttack* approach with the out-of-topic cross-domain setting.

Dataset	Model	Ori $F1_{macro}$	Att $F1_{macro}$	Success Rate	Query Number
P-STANCE	BART	0.826	0.142	83.8%	15.4
	BERT	0.768	0.150	86.6%	15.3

Table 14: Results on P-STANCE dataset. Lower Att $F1_{macro}$ Scores Indicate Better Performance.

Dataset	Model	Success Rate (%)	Avg. Queries
Covid	BART	99.2	8.5
	Gemini	94.97	12.98
VAST	BART	76.6	40.2
	Gemini	75.4	41.7

Table 15: Attack performance of StanceAttack across different victim models. Results show that StanceAttack remains effective when targeting both PLMs (BART) and larger LLM-based models (Gemini 2.0 Pro).

K LLM as Victim Models

To further evaluate the effectiveness of StanceAttack on LLM models, we extend our experiments to include a state-of-the-art LLM (Gemini 2.0 Pro) as the target model. The results are shown in Table 15.

We observe that StanceAttack achieves consistently high success rates across both PLMs and LLM-based victims. Specifically, on the Covid dataset, the attack reaches a 94.97% success rate against Gemini 2.0 Pro, compared to 99.20% on BART. On VAST, the success rate remains comparable (75.40% vs. 76.60%). While attacking stronger LLMs requires slightly more queries, the overall effectiveness remains largely preserved.

These results demonstrate that STANCEATTACK generalizes well to more capable LLM models and is not limited to small or medium-scale PLMs, supporting its applicability in more realistic and modern deployment scenarios.

L Human evaluation on Stance Preserving

To evaluate whether our paraphrasing-based adversarial attack preserves stance orientation, we randomly select 100 examples and have three human annotators assess whether the rephrased text retains the same stance toward the target as the original. Results show that 93% of rephrased texts maintain their stance. Detailed annotation procedures are provided in Appendix M.

M Annotation Details

Human evaluation results are gathered through Cogitotech,[4] a company specializing in data annotation for major AI organizations (e.g., OpenAI, AWS, etc.). To guarantee the quality of annotations, we adhere to stringent criteria: 1) Annotators are required to have at least a college degree; 2) Annotators must be native English speakers. Additionally, we conduct quality checks by randomly reviewing 10% of each annotator's work and exclude any annotator whose acceptance rate falls below 90%. Annotations that do not meet our standards are reassigned to other qualified annotators for reevaluation.

We present annotators with the original text (Text A), the adversarial text (Text B), and the target. with the following instructions: *You will be provided with two texts: Text A and Text B. Please determine whether Text B exhibit similar semantic meanings as Text A (select between: Yes or No). Please rate the grammatical correctness of Text B on a scale from 1 to 5. The lower the score, the more grammatical errors you observed. Please annotate whether the Text A and Text B express the same stance towards the target (select between: Yes or No).*

We calculated the Krippendorff's alpha score for our human annotators, which are 0.73 and 0.8 for semantic and stance preserving annotation tasks, respectively.

[4] https://www.cogitotech.com/