Adversarial Style Augmentation via Large Language Model for Robust Fake News Detection

Sungwon Park*
KAIST
Daejeon, South Korea
psw0416@kaist.ac.kr

Sungwon Han*
KAIST
Daejeon, South Korea
lion4151@kaist.ac.kr

Xing Xie
Microsoft Research Asia
Beijing, China
xingx@microsoft.com

Jae-Gil Lee
KAIST
Daejeon, South Korea
jaegil@kaist.ac.kr

Meeyoung Cha
MPI-SP & KAIST
Bochum, Germany
meeyoungcha@kaist.ac.kr

Abstract

The spread of fake news harms individuals and presents a critical social challenge that must be addressed. Although numerous algorithmic and insightful features have been developed to detect fake news, many of these features can be manipulated with style-conversion attacks, especially with the emergence of advanced language models, making it more difficult to differentiate from genuine news. This study proposes adversarial style augmentation, AdStyle, designed to train a fake news detector that remains robust against various style-conversion attacks. The primary mechanism involves the strategic use of LLMs to automatically generate a diverse and coherent array of style-conversion attack prompts, enhancing the generation of particularly challenging prompts for the detector. Experiments indicate that our augmentation strategy significantly improves robustness and detection performance when evaluated on fake news benchmark datasets.

CCS Concepts

• **Security and privacy → Software and application security**;
• **Computing methodologies → Artificial intelligence**.

Keywords

Misinformation, Adversarial Training, Large Language Model

ACM Reference Format:
Sungwon Park, Sungwon Han, Xing Xie, Jae-Gil Lee, and Meeyoung Cha. 2025. Adversarial Style Augmentation via Large Language Model for Robust Fake News Detection. In *Proceedings of the ACM Web Conference 2025 (WWW '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3696410.3714569

*Equal contribution to this work

WWW '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1274-6/25/04
https://doi.org/10.1145/3696410.3714569

1 Introduction

In today's digital landscape, people seek information through various channels, including social media. News content continues to hold substantial importance on these platforms, significantly shaping individuals' thoughts and decisions. However, the ease of sharing and consuming information outside traditional news outlets has introduced several challenges. Among them is the rise of alternative media and news-like content, which can contain completely or partially false information [28]. The spread of such misinformation has negative impacts on individuals and is considered a major social challenge that requires urgent attention [5].

Given the vast amount of information shared on social platforms, manually detecting fake news has become impractical. Consequently, these efforts are increasingly supported and managed by algorithms. For example, sentiment and topic features, as well as the temporal and structural patterns of diffusion networks, have proven effective in detecting fake news [14, 22, 23]. These patterns can be utilized by machine learning and deep learning [17, 21, 26]. Large language models (LLMs) have also been used to learn the textual style of fake news by extracting sentence embeddings to train detectors [4, 7, 11].

Although certain textual styles or linguistic cues can indicate fake news, attackers can circumvent such detection by rearranging the order of subjects and objects or by imitating specific styles [13, 35]. LLMs can easily paraphrase sentences in any desired manner through prompts (e.g., "change this text into a NYTimes style"), making it challenging to differentiate AI-generated news from authentic ones. These manipulations, referred to as style-conversion attacks, significantly complicate the task of maintaining information veracity in the digital media landscape [13, 35].

In this study, we present AdStyle, an adversarial style augmentation method designed to train a fake news detector that can withstand various style-conversion attacks. Unlike previous methods that relied on predefined style-conversion prompts, our approach identifies prompts to generate adversarial style augmentations tailored to a specific detector. This process adds noise to the style features within the detector's decision boundary, while preserving content-wise integrity. To customize prompts that are adversarial to the detector, we implement an automated prompt engineering technique, similar to [33]. By providing style-conversion prompts and evaluating the detector's performance under these augmentations,

the LLM can discern patterns between prompts and performance, facilitating search for the most effective prompts.

When evaluated on fake news benchmarks such as PolitiFact, GossipCop, and Constraint under various style-conversion attack scenarios, our augmentation strategy exhibited enhanced robustness and detection performance compared to current state-of-the-art methods. Our strategy preserves the content of the sentence while altering its structure to increase perplexity according to the LLM-based detector. This adjustment results in a sentence structure that the LLM has not frequently encountered during its pre-training phase, effectively deterring malicious style-conversion attacks, such as paraphrasing. Additionally, our method serves as an augmentation strategy that can be integrated with any existing detection models, regardless of their design.

We release the code and implementation details of our fake news detection against style-conversion attacks for the broader use and greater impact within the research community and industry: https://github.com/deu30303/AdStyle.

2 Related Work

2.1 Automated Detection of Fake News

Among the machine-driven fake news detection methods are learning methodologies extract textual features from fake news texts using benchmark datasets and ground-truth labels [25]. These textual features can include deep features based on artificial neural networks [26] as well as manually defined features such as sentiment and political bias [14, 22, 23]. Other approaches include domain adaptation methods to enhance detection generalizability across various domains and topics [19, 20], and knowledge-based methods that rely on external data to distinguish false information [9]. With the advent of LLMs, new methods have emerged that utilize LLMs' prior knowledge to identify fake news [11], including research on detecting machine-generated fake news [7]. One approach involves using LLMs to generate synthetic reactions and comments on news articles from diverse perspectives, which are then leveraged for fake news detection [29]. Furthermore, LLM-generated logic has been employed to enhance the interpretability of fake news detection models [16]. Despite these advancements, fake news detection using LLMs remains a challenging task due to factuality hallucinations, which can undermine reliability [3].

2.2 Attack on Fake News Detection

Various attack methods have been studied to test the robustness of fake news detection [13, 35]. These methods include injecting misinformation by rearranging subjects and objects or causes and effects while retaining the textual features used for detection [13]. Other techniques involve creating fact distortions by modifying or exaggerating words related to people, time, or places, while preserving sentence structure [35]. Another line of research simulates the behavior of malicious users on social media by manipulating news article comments or employing a multi-agent reinforcement learning framework to generate adversarial content [15, 30]. Recently, methods leveraging the text generation capabilities of LLMs to modify the writing style of sentences have also been proposed [31]. In this study, we propose an adversarial training method to mitigate style-conversion attacks in fake news detection. Our approach utilizes automated prompt engineering to identify the most effective style-conversion prompts for a given fake news dataset [33, 34].

3 Method

3.1 Overview

Let $\mathcal{D} = \{(\mathbf{d}_i, y_i)\}_{i=1}^{N}$ be a dataset containing news $\mathbf{d}_i$ and the corresponding ground-truth binary veracity label y_i (indicating whether the news is true or fake). Each news item $\mathbf{d}_i$ is composed of natural language-based text. Our goal is to train a fake news detector f based on the labeled dataset and predict veracity labels. In doing so, we ensure that a detector remains robust even when an attacker perturbs the textual style, such as the order and format, while preserving the meaning of the sentences.

Figure 1 illustrates the workflow. Our method, called AdStyle, generates style-conversion prompts and performs augmentation over multiple rounds of the following process. Firstly, leveraging the reasoning ability of LLM, adversarial style-conversion prompt candidates are generated. These style-conversion prompts contain instructions on how to transform the given sentences and are used as inputs to the LLM along with the original sentences for conversion. Following the automated prompt engineering [33], the style-conversion prompts and detector prediction score pairs used in previous rounds are included as in-context demonstrations to guide the LLM in searching prompt candidates that maximally confuse the detector (Section 3.2). Next, a subset of the dataset is selected, and these discovered candidates are applied to perform conversions. The converted samples are then evaluated to determine how much they confuse the detector. From these candidates, the top-k prompts that are diverse and maintain the original content's meaning while most effectively confusing the detector are selected (Section 3.3). These selected prompts are used as an augmentation method to train the detection model in the current round. We describe each step's details below.

3.2 Adversarial Style-Conversion Prompts

The style-conversion prompts we generate are instructions that perturb only the textual style, such as the structure or format of the sentence, while preserving the content of the given input text. For example, an instruction like "Rewrite the following article in an objective and professional tone" can transform the stylistic features of fake news to resemble authentic news. However, if these instructions are heuristically defined, they may not align with the detector's actual decision boundary, thereby reducing training efficiency. Furthermore, if the instructions remain fixed throughout the training process, the detector may memorize these conversion patterns, leading to overfitting.

Instead of predefining and fixing the instructions for conversion, we identify adversarial conversion prompts that introduce noise directed toward the decision boundary of the current detector, thereby maximizing prediction confusion. This approach is similar to adversarial training commonly used in the computer vision domain [18], which enhances robustness against slight input perturbations. However, for textual data, the discrete nature of the input makes it challenging to add noise directly through the detector's gradient as in

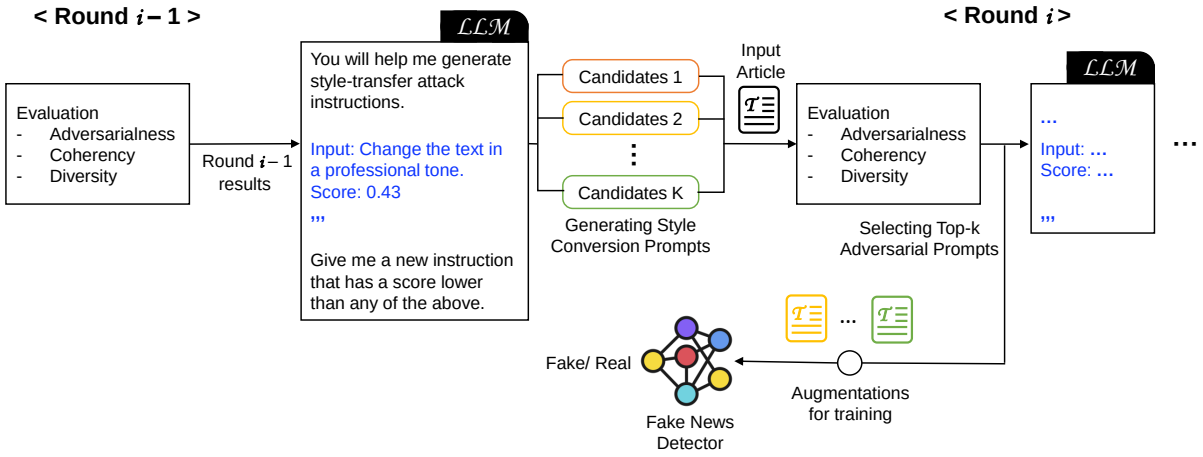


Figure 1: Illustration of AdStyle. The model undergoes multiple rounds of training. In each round, it utilizes the style-conversion prompt and prediction confusion score from the previous round to create prompt candidates aimed at maximizing the detector model's confusion. Subsequently, a subset of the training dataset is used to select the top-k prompts, based on factors such as adversarialness, coherency, and diversity. These selected prompts are then employed as augmentations in the training process. Repetition of this process allows the model to consider a wide range of adversarial prompts, making the director more resilient to future attacks.

> Now you will help me minimize a fake news detector performance value with style transfer attack instructions. I have a set of style transfer attack instructions that I will use to minimize the fake news detector's performance. The pairs are arranged in descending order based on their function values, where lower values are better.
>
> Input: [Style-Conversion Prompt Example]
> Score: [Prediction Confusion Score]
>
> Input: ... ...
>
> Give me a new style transfer attack instruction that is different from all pairs above and has a lower performance value.

Figure 2: Prompt for generating adversarial style-conversion prompts. The text in blue represents the score trajectory, and the remaining text represents the problem description.

the image domain. To address this, we measure prediction confusion using conversion prompts (i.e., prediction confusion score) and feed this prompt-score pair to the LLM for generating adversarial prompts, inspired by automated prompt engineering techniques.

Figure 2 shows an example of the LLM input used to generate adversarial style-conversion prompts. The LLM input consists of the problem description and the score trajectory components. We describe each component in turn.

Problem description component. This component includes the problem description, the objective, and constraints on the response necessary for generating style-conversion prompts. For example, a sentence like "Minimize a fake news detector performance value with style transfer attack instruction" informs the LLM of the intent behind the conversion prompt.

Score trajectory component. Previous research has demonstrated that LLMs can learn patterns from in-context demonstrations provided as input [8, 33]. The score trajectory component leverages this ability by providing style-conversion prompts from previous rounds and their corresponding prediction confusion scores in the form of in-context demonstrations. In the first round, a predefined set of prompts is excerpted from [31]. The score is calculated by selecting a subset $\mathcal{B}$ from the entire training dataset $\mathcal{D} = \{(\mathbf{d}_i, y_i)\}_{i=1}^{N}$, applying a conversion prompt c to create a new set $\mathcal{B}^c = \{(\mathbf{d}_i^c, y_i)\}_{i=1}^{M}$, where $\mathbf{d}_i^c = \text{Convert}(c, \mathbf{d}_i)$, $N >> M$, and then measuring the AUC score between the predictions and the ground-truth labels when $\mathcal{B}^c$ is fed into the detector. Specifically, the score of a conversion prompt s_c is defined as:

$$s_c = |0.5 - \text{AUC}(\{y_i\}_{i=1}^{M}, \{f(\mathbf{d}_i^c)\}_{i=1}^{M})|. \quad \text{(Eq. 1)}$$

A lower score indicates a higher level of prediction confusion, implying that the conversion prompt has caused the detector's predictions to become random with respect to the labels. Finding conversion prompts that lead to high confusion (i.e., low score) suggests that the detector has not yet learned to handle those stylistic features, and the conversion prompts have placed the samples near the detector's decision boundary, making them difficult to distinguish. By providing these in-context demonstrations, the LLM can generate conversion prompts that differ from previous ones while maximizing confusion (i.e., minimizing the score). It is important to avoid selecting conversion prompts that flip the detector's original predictions (i.e., AUC << 0.5), as such cases will disrupt the content and stylistic features. We extract S style-conversion prompts at a time using the input described above.

3.3 Selecting Top-k Adversarial Prompts

Using all the style-conversion prompt candidates generated by the LLM can be computationally intensive, and not all prompts may be suitable for augmentation. For example, the set of style-conversion prompts used for augmentation should each provide adversarial perturbations that confuse the detector (i.e., adversarialness). While altering the textual style, the prompts should not change the meaning of the sentences to prevent label noise (i.e., coherency). The more diverse the set of conversion directions covered by the prompts, the more efficient the augmentation process (i.e., diversity).

To select a set of style-conversion prompts that satisfies these three criteria, we propose a selection strategy. Given a conversion prompt c, we first extract embedding vectors for the input texts in both the training subset $\mathcal{B}$ and the subset $\mathcal{B}^c$ converted by c using a large language model g such as BERT. We then compute the average embedding vectors for each subset and calculate the vector difference $\mathbf{z}_c$.

$$\mathbf{z} = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{d}_i \in \mathcal{B}} g(\mathbf{d}_i), \quad \mathbf{z}' = \frac{1}{|\mathcal{B}'|} \sum_{\mathbf{d}_i^c \in \mathcal{B}'} g(\mathbf{d}_i^c)$$
$$\mathbf{z}_c = \mathbf{z}' - \mathbf{z} \qquad \text{(Eq. 2)}$$

Here, $\mathbf{z}_c$ represents the average change in embedding direction due to the conversion prompt c. We then calculate the adversarialness scale s^c_{adv} and coherency scale s^c_{coh} for each conversion prompt c, and rescale $\mathbf{z}_c$ accordingly (i.e., $\hat{\mathbf{z}}_c = \mathbf{z}_c \times s^c_{\text{adv}} \times s^c_{\text{coh}}$). Finally, we use the k-means++ initialization method [1] on these rescaled vectors to select k prompts. The k-means++ initialization method helps select a diverse set of prompts that are adversarial and coherent, by choosing samples that are as far apart as possible [2]. The details of each scale are described below.

Adversarialness scale (s^c_{adv}). To measure how adversarial a given style-conversion prompt is to the detector, we newly define the adversarialness scale similar to the confusion score defined in the previous section. Given the converted batch $\mathcal{B}^c$ by conversion prompt c, the adversarialness scale s^c_{adv} is defined as:

$$s^c_{\text{adv}} = -1.8 \cdot |\text{AUC}(\{y_i\}_{i=1}^M, \{f(\mathbf{d}_i^c)\}_{i=1}^M) - 0.5| + 1. \qquad \text{(Eq. 3)}$$

This value increases as the AUC approaches 0.5, indicating that the prediction is more random. The coefficient 1.8 ensures that the scale ranges between 0.1 and 1, preventing it from being zero.

Coherency scale (s^c_{coh}). To verify that the converted text retains the same content as the original text, we check the similarity in meaning between the text pairs using an LLM. We create sample pairs from $\mathcal{B}$ and the converted subset $\mathcal{B}^c$, and inquire the LLM about the percentage of pairs that it considers to have the same meaning. This percentage is used as the coherency scale s^c_{coh}. Like the adversarialness scale, this value is rescaled to range between 0.1 and 1.

Finally, the selected style-conversion prompts via our score and k-means++ initialization method are applied to the input texts of the entire dataset $\mathcal{D}$ to create augmented samples. These augmented samples are then used alongside the original samples to train the detector f. The detector is trained using binary cross-entropy loss. These prompt generation and selection processes are repeated over multiple training rounds.

4 Experiment

We evaluate the robustness of AdStyle under diverse style-conversion attacks across multiple datasets, comparing it with contemporary baselines. We then analyze the impact of various model components on overall performance. A comprehensive evaluation is also performed on a broader range of paraphrasing attacks, including comparisons with LLM-based zero-shot and in-context learning baselines. Lastly, a qualitative analysis is conducted to explore the characteristics of the generated style-conversion prompts.

4.1 Performance Evaluation

Dataset. We use three real-world fake news benchmark datasets: (1) PolitiFact and (2) Gossipcop, released by FakeNewsNet benchmark [27], which focus on political claims and celebrity rumors, respectively, and (3) Constraint [10], a dataset of social media posts on COVID-19. Datasets were split into an 80% training set and a 20% test set, as described in Table 1.

Table 1: Statistics of fake news datasets.

Dataset	PolitiFact	GossipCop	Constraint
# of News Articles	774	7,916	8,418
# of Real News	399	3,958	4,406
# of Fake News	375	3,958	4,012

Attack settings. To assess robustness against style conversion attacks, we employ LLM-empowered techniques to reframe the test set using a variety of style conversion prompts, as illustrated in Figure 3. Following the original literature [31], we use four prominent daily news sources as [publisher name]: **CNN**, **The New York Times**, **The Sun**, and **National Enquirer**. CNN and The New York Times were chosen as representative news outlets recognized for their reputable journalism, while The Sun and National Enquirer are characterized by their tabloid style. We utilize OpenAI's GPT-3.5-Turbo model for reframing input sentences, with the temperature set to 0 and the top-p value set to 1 by default. Experiments are conducted using 10%, 25%, and 100% of the complete dataset to observe the impact of augmentation on training performance, measured by AUC, across various dataset sizes.

> Rewrite the following article using the style of [publisher name]: [news article]

Figure 3: Prompt for style conversion. The "publisher name" part will be filled with the name of a representative publisher (e.g., newspaper or journal), and the "news article" part will contain the original news text.

Baselines. We implemented several existing text-based fake news detection strategies as baselines: (1) Vanilla, a text-based fake news detector using conventional binary cross entropy objective; (2) UDA, introduces consistency regularization objective between original text and diverse augmented variations. [32]; (3) RADAR, utilizes an

Table 2: Performance comparison with AdStyle in two different scenarios—Attack, where style-conversion attacks are performed, and Clean, where no attack is performed—across three fake news datasets. For the attack scenario, we report the average AUC of four style-conversion attacks. The best results are marked in bold. The values 0.1, 0.25, and 1 indicate the proportion of the dataset used for training relative to the full dataset. Our model demonstrates a significant performance improvement over all text-based fake news detectors in both style-conversion attack and clean scenarios.

Attack	Politifact			Gossipcop			Constraint		
	0.1	0.25	1	0.1	0.25	1	0.1	0.25	1
Vanilla	0.6114	0.6904	0.8548	0.6920	0.7876	0.8453	0.8185	0.8674	0.8741
UDA	0.6241	0.7696	0.8564	0.7381	0.7865	0.8591	0.8615	0.9028	0.9297
RADAR	0.7218	0.7399	0.8571	0.7583	0.8170	0.8616	0.8535	0.8047	0.9086
ENDEF	0.6376	0.7579	0.8134	0.7405	0.7870	0.8615	0.8234	0.8950	0.8835
SheepDog	0.6525	0.8234	0.9009	0.7498	0.8357	0.8669	0.8926	0.9188	0.9630
AdStyle	**0.7833**	**0.8919**	**0.9399**	**0.8134**	**0.8389**	**0.8721**	**0.9224**	**0.9531**	**0.9716**

Clean	Politifact			Gossipcop			Constraint		
	0.1	0.25	1	0.1	0.25	1	0.1	0.25	1
Vanilla	0.7393	0.8397	0.9355	0.7096	0.8104	0.8645	0.9311	0.9682	0.9892
UDA	0.7404	0.8783	0.9422	0.7422	0.8022	0.8666	0.9365	0.9724	**0.9899**
RADAR	0.7607	0.8495	0.9314	0.7593	0.8170	0.8630	0.9446	0.9773	0.9817
ENDEF	0.7776	0.8823	0.9294	0.7592	0.7991	0.8738	0.9234	0.9556	0.9871
SheepDog	0.7248	0.8229	0.9394	0.7490	0.8411	0.8641	0.9144	0.9459	0.9785
AdStyle	**0.8996**	**0.9280**	**0.9460**	**0.8251**	**0.8493**	**0.8797**	**0.9509**	**0.9849**	0.9889

Table 3: Performance comparison of ablations on the Politifact dataset. The results show the impact of style-conversion attacks using four different publishers (i.e., CNN, The New York Times, The Sun, and National Enquirer) and a clean scenario (i.e., Clean) where no attack is performed. Any modification or removal of model components leads to decreased performance.

Model	CNN	The New York Times	The Sun	National Enquirer	Clean
Vanilla	0.8127	0.8687	0.8789	0.8591	0.9355
Random Selection	0.9075	0.9409	0.9355	0.9365	0.9412
Class Prompt	0.9131	0.9272	0.9333	0.9313	0.9471
Adversarial only Selection	0.9035	0.9343	0.9318	0.9288	0.9405
w/o Adversarialness	0.9039	0.9333	0.9320	0.9402	**0.9473**
w/o Coherency	0.9193	0.9315	0.9297	0.9430	0.9409
w/o Score trajectory	0.8199	0.8917	0.9124	0.9066	0.9372
Full Components	**0.9174**	**0.9444**	**0.9520**	**0.9460**	0.9460

adversarially trained paraphraser to generate augmented version of input sentences [12]; (4) ENDEF, mitigates entity bias in fake news data through causal learning [36]; (5) SheepDog, introduces predefined style-conversion prompts to augment the styles of input text via LLM [31]. We follow the original paper's setting and details for baseline implementations.

Implementation details. All models are evaluated under uniform experimental conditions to ensure fair comparison. This consistency extends to the choice of backbone network, optimizer, and learning rate. We utilize OpenAI's GPT-3.5-Turbo model for reframing input sentences and measuring coherency, with the temperature set to 0 and the top-p value set to 1 by default. The training process for the detector is conducted over 10 rounds, with one training epoch per round. When evaluating the style-conversion prompts, we randomly selected 30 samples from the training dataset to apply augmentation (i.e., $M = 30$). In each round, 30 prompt candidates were generated by the LLM (i.e., $S = 30$), from which 3 were chosen for augmentation (i.e., $k = 3$) by our selection strategy. The training utilizes the AdamW optimizer with a learning rate of 1e-5 and a batch size of 8. For measuring diversity, the text embeddings are generated using a pre-trained BERT-based uncased model from HuggingFace Transformers. Two V100 GPUs were utilized for all experiments.

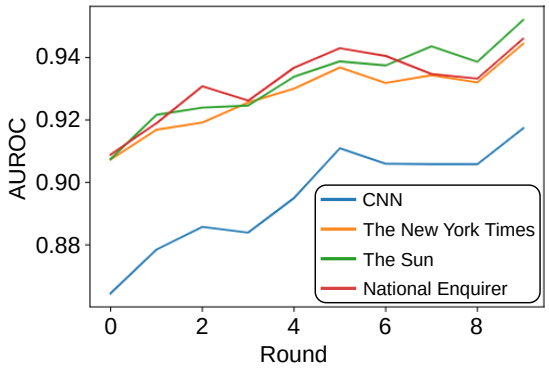


Figure 4: Performance changes across rounds on the PolitiFact dataset for four different style-conversion attacks. The x-axis represents the training rounds, and the y-axis represents the detector's AUC. For all attacks, the detector's performance improved as the rounds progressed.

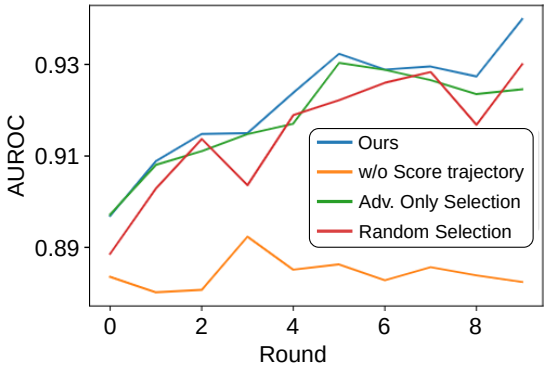


Figure 5: Performance of selection strategies over training rounds on PolitiFact. The x-axis represents the training rounds, while the y-axis represents the detector's AUC.

Table 4: Performance comparison of different LLM-based baselines on the PolitiFact Dataset. (NY: The New York Times, TS: The Sun, NE: National Enquirer)

Model	CNN	NY	TS	NE	Clean
GPT-3.5 zero-shot	0.5820	0.6242	0.6173	0.5274	0.7037
GPT-3.5 in-context	0.6954	0.6504	0.6875	0.5754	0.7383
Ours	0.9174	0.9444	0.9520	0.9460	0.9460

Result. Table 2 shows results comparing the performance of detector algorithms in both clean scenarios, where no attack is performed, and adversarial scenarios, where style-conversion attacks are applied. Due to space limitations, the average AUC results under the four different style-conversion attacks are reported. We find that AdStyle consistently outperforms the baselines, including both clean and adversarial scenarios. This demonstrates that our augmentation strategy enhances both the robustness and generalizability of the detector. The model is particularly effective when compared to other baselines and for smaller datasets. Figure 4 shows the AUC for each style-conversion attack over different rounds. The performance gradually improves as the rounds progress, indicating that continually discovering adversarial augmentations is beneficial.

4.2 Component Analysis

We now examine the contribution of each component on our adversarial style-conversion prompts. The proposed method integrates two main modules: adversarial style-conversion prompts generation and selection. To evaluate their individual contributions, we conduct experiments where we either remove each component or substituted it with an alternative within the full model. This results in six distinct configurations for analysis: (1) **Full Components**: Our complete method with all components; (2) **Random Selection**: The method that randomly select conversion prompts from candidates instead of using our selection strategy (Sec. 3.3); (3) **Class Prompt**: The method categorizes the confusion scores of conversion prompts into three levels: high, medium, or low. These categorized labels are then used as in-context demonstrations for generating adversarial style-conversion prompts (Sec. 3.2), instead of relying on continuous confusion scores s_c (Eq. 1); (4) **Adversarial only Selection**: The method that selects top-k adversarial prompts, not considering diversity and coherence criteria; (5) **w/o Adversarialness**: The method that omits the adversarialness criterion in our selection strategy; (6) **w/o Coherency**: The method that omits the coherency criterion in our selection strategy; (7) **w/o Score trajectory**: The method without score trajectory component for the style-conversion prompt generation (Sec. 3.2).

Table 3 demonstrates that omitting any component leads to a decrease in performance for certain style conversions attack. Notably, excluding the score trajectory component when selecting the style-conversion prompt proved to be the most detrimental to performance. This finding suggests that the LLM can effectively identify conversion prompts that may confuse the detector through the style-conversion prompt and confusion score pairs as in-context demonstrations. Moreover, selectively choosing conversion prompts identified by the LLM based on specific criteria resulted in further performance improvements. Our sampling strategy also facilitates faster convergence compared to alternative sampling methods (See Figure 5).

4.3 Performance Analysis

We here conduct analysis on how AdStyle demonstrates robust and high performance across various scenarios and how it effectively enhances the detector's performance.

Comparison with LLM-based baselines. AdStyle enhanced the detector's performance by leveraging the reasoning abilities of advanced large language models like GPT-3.5. To determine whether the capabilities of an advanced large language model alone are sufficient for the fake news detection task, we compared AdStyle with other LLM-based baselines: zero-shot and in-context learning-based inference with GPT-3.5. In the case of the in-context learning baseline, one example each of fake news and real news was provided

Table 5: Comparison under attack scenarios with Gemini-Pro on the PolitiFact dataset.

Model	CNN	NY	TS	NE	Average
Vanilla	0.7913	0.8039	0.8882	0.8397	0.8308
UDA	0.7216	0.7863	0.8657	0.8119	0.7964
RADAR	0.8023	0.7805	0.8764	0.8423	0.8254
ENDEF	0.7730	0.7537	0.8439	0.8058	0.7941
SheepDog	0.8487	0.8926	0.9174	0.8998	0.8896
Ours	0.8821	0.9120	0.9295	0.9241	0.9120

as in-context demonstrations. The example instruction prompt for the LLM-based baselines are as follows:

Table 4 presents the comparison results on the PolitiFact dataset. Using LLMs with prompting alone (Figure 6) makes it challenging to accurately determine the authenticity of news. Instead of 'directly' leveraging the LLM's text generation capability for inference, it is more effective to use it as an augmentation tool to provide additional training signals, as demonstrated by our model.

> Does the following contain real or fake news? Answer in one word with either 'Real'or 'Fake': [news article]

Figure 6: Instruction prompt for LLM-based baselines. The "news article" section contains the original news text.

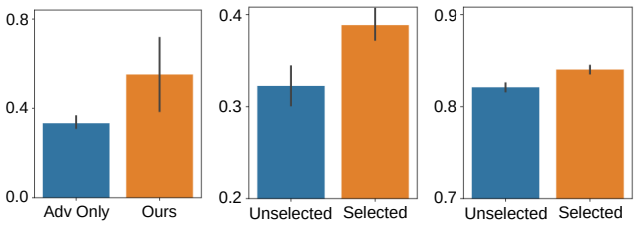
Robustness against a other LLM backbones. We assessed the robustness by examining its performance against style conversion attacks when the attacker utilizes a different LLM backbone, such as Gemini-Pro [24]. As shown in Table 5, AdStyle consistently outperforms the baselines even with a different LLM backbone, suggesting that the proposed approach is not overly reliant on recognizing the specific style of content generated by the backbone.

Robustness against other possible attack scenarios. To validate the model's robustness against various attacks, we have conducted comparison experiments with diverse attack scenarios to deceive the detector by altering textual style of inputs: (1) Adversarial prompt: Given a news article and its label, the prompt instructs the LLM to rewrite the article to evade detection as the given label. (2) Summarization prompt: A prompt instructing the LLM to summarize the news article without incorporating stylistic guidance. (3) In-Context prompt: A prompt providing an example of a recent, real CNN article, instructing the LLM to rewrite the given article in the same style. (4) Adversarial Paraphraser: The attack utilizes a paraphraser adversarially trained on the training dataset. The example prompt for each attack is described in Figures 12 to 14 in Appendix. Based on the results in Table 6, we confirm that our proposed model still demonstrates a performance improvement compared to other baselines against these attacks.

Effectiveness of the selection strategy. We here empirically verified that our selection strategy (Sec. 3.3) effectively selects diverse style-conversion prompts with high adversarialness and coherency.

Table 6: Comparison under attack scenarios on the PolitiFact Dataset (A: Adversarial prompt, B: Summarization prompt, C: In-Context prompt D: Adversarial Paraphraser).

Model	A	B	C	D
Vanilla	0.8881	0.8522	0.8896	0.8305
UDA	0.8624	0.8363	0.8924	0.8682
RADAR	0.9096	0.8754	0.9234	0.9007
ENDEF	0.8628	0.7978	0.9009	0.8682
SheepDog	0.9297	0.9205	0.9276	0.8995
Ours	0.9456	0.9416	0.9425	0.9212



(a) Diversity **(b) Adversarialness** **(c) Coherency**

Figure 7: Qualitative analyses with y-axis representing (a) diversity in embedding (z_c) of conversion prompts sampled by AdStyle and adversarial-only selection; (b) Adversarialness (s^c_{adv}) for selected and unselected prompts; and (c) Coherency for selected and unselected prompts.

Figure 7a visualizes the diversity of prompts selected by our method compared to those chosen solely based on adversarialness scores (i.e., Adversarial-only Selection, the third model in our ablation study). We measure diversity using the average cosine similarity of every pair of selected prompts' embeddings, $\mathbf{z}_c$ (Eq. 2):

$$\text{Diveristy}(C) = 1 - \frac{1}{|C|} \sum_{(c_i,c_j)\in C} \text{sim}(\mathbf{z}_{c_i}, \mathbf{z}_{c_j}), \qquad \text{(Eq. 4)}$$

where C is the set of prompt pairs, $\text{sim}(\cdot)$ represents the cosine similarity. The result in the Figure 7a indicate that our strategy results in a more diverse set of augmentations compared to selection based on adversarialness alone.

Figure 7b and 7c illustrate the Adversarialness and Coherency, respectively, of our selected prompts compared to remaining unselected prompts. Adversarialness was measured using the s^c_{adv} score (Section 3.3), and coherency was calculated as the cosine similarity between the original text and its augmented version using semantic BERT embeddings [6]. We can also observe that, for both metrics, prompts selected through AdStyle exhibit higher values compared to unselected prompts. This suggests that our strategy effectively selects prompts by considering both adversarialness and coherency.

Analysis on style-conversion prompts. Finally, we conducted a qualitative analysis of our style-conversion prompts to understand what characteristics of the augmented samples contribute to improving the detector's robustness. Interestingly, when we compare the augmented samples from two models: AdStyle and SheepDog,

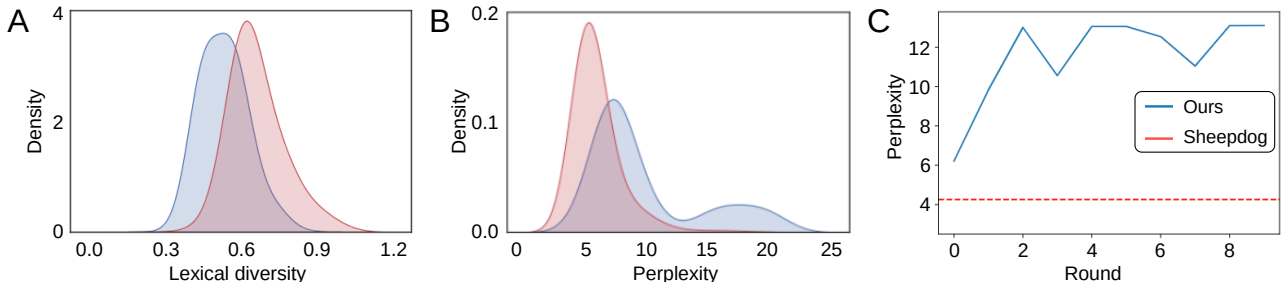


Figure 8: Comparison of augmented samples generated by our model and SheepDog: (A) Histogram of lexical diversity, (B) Histogram of perplexity, and (C) Perplexity across rounds.

Round 0
P1: Rewrite the following article in a nonsensical and absurdly exaggerated tone with a hint of horror
P2: Rewrite the following article in a sarcastic and mocking tone
P3: Rewrite the following article in a chaotic and disorganized tone
Round 1
P1: Rewrite the following article in a haunting and macabre tone with a sense of impending horror and madness
P2: Rewrite the following article in a cryptic and enigmatic tone
P3: Rewrite the following article in a malevolent and apocalyptic tone with a sense of impending doom and destruction, while also incorporating elements of surrealism and existential dread

Figure 9: Example adversarial style-conversion prompts selected for the PolitiFact dataset.

we found that our model produced sentences with significantly higher perplexity according to the language model backbone used by the detector than SheepDog model (9.05 vs. 4.26, see Figure 8B), even though ours have lower lexical diversity (0.503 vs. 0.634, see Figure 8A). In other words, our generated samples featured sentence structures that the detector's language model likely had not encountered during pretraining. By providing the detector with inputs that have a variety of sentence structures and styles it has not previously seen, the detector naturally becomes robust against style-conversion attacks. In addition, when we measured the changes in perplexity of the augmented samples across training rounds (see Figure 8C), the perplexity increased over rounds and eventually converged at a certain point. This indicates that iterative exploration over multiple rounds is effective in creating augmentations that increasingly challenge the detector.

Figure 9 shows examples of prompts generated and selected by AdStyle. In contrast, our prompts included creative phrases beyond typical human-crafted suggestions, making them adversarial to the detector. The round 0 result shows a prompt asking for "(P1) a nonsensical and absurdly exaggerated tone with a hint of horror", while round 1 offers a lengthier prompt mentioning, for example, "(P3) a malevolent and apocalyptic tone with a sense of impending doom and destruction, while also incorporating elements of surrealism and existential dread." Previous work relied on conventional prompts, such as "neutral" or "sensational." The LLM is particularly well-suited for exploring such a vast array of candidates. Currently, we use a fixed format for the initial set of prompts, which results in the generation of prompts with a similar format. We expect that using a broader set of prompts will allow the LLM to optimize prompts within a wider search space.

5 Conclusion

We presented a robust fake news detection method that effectively withstands paraphrasing attacks through adversarial style conversion. Unlike traditional detectors that use predefined and agnostic augmentations, AdStyle employs tailored augmentations that shift samples in the direction of the detector's current decision boundary using style-conversion prompts, functioning similarly to adversarial noise. Among the various candidates for the LLM prompts, we selected an efficient set of prompts for training by considering diversity, coherence, and adversarialness.

Consequently, we were able to train a detector that exhibits high robustness and generalizability against a wide range of attacks. This robustness ensures that the detector can effectively identify and mitigate various forms of style-conversion attacks, regardless of the target news outlet. We believe that our research, combined with the shared codes, contributes to better filtering of fake news online. We aim to support the broader goal of ensuring the integrity of information in the digital age.

Ethical Consideration

We recognize several ethical concerns in this work. First, our method may inherit biases from the training data, which can be mitigated by carefully selecting style-conversion results that preserve textual coherence. Second, while our approach aims to detect fake news, similar techniques could be misused to enhance misinformation—an ongoing challenge in adversarial research. Lastly, our detection methods might influence news production by constraining creative writing styles, highlighting the need for further research on the balance between algorithmic moderation and information systems.

Acknowledgments

J.G. Lee and M. Cha are co-corresponding authors. We extend our gratitude to Fangzhao Wu, Wenchao Dong, and the anonymous reviewers for their insightful feedback on our work. This research was supported by the National Research Foundation of Korea (NRF) grant (RS-2022-00165347).

References

[1] David Arthur, Sergei Vassilvitskii, et al. 2007. k-means++: The advantages of careful seeding. In *Soda*, Vol. 7. 1027–1035.

[2] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *Proc. of International Conference on Learning Representations*.

[3] Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, et al. 2024. Factuality challenges in the era of large language models and opportunities for fact-checking. *Nature Machine Intelligence* 6, 8 (2024), 852–863.

[4] Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, Eduard Hovy, Heng Ji, Filippo Menczer, Ruben Miguez, Preslav Nakov, Dietram Scheufele, Shivam Sharma, and Giovanni Zagni. 2024. Factuality challenges in the era of large language models and opportunities for fact-checking. *Nature Machine Intelligence* (2024).

[5] Meeyoung Cha, Chiyoung Cha, Karandeep Singh, Gabriel Lima, Yong-Yeol Ahn, Juhi Kulshrestha, Onur Varol, et al. 2021. Prevalence of misinformation and factchecks on the COVID-19 pandemic in 35 countries: Observational infodemiology study. *JMIR human factors* 8, 1 (2021), e23279.

[6] Sachin Chanchani and Ruihong Huang. 2023. Composition-contrastive Learning for Sentence Embeddings. In *Proc. of Association for Computational Linguistics*. 15836–15848.

[7] Canyu Chen and Kai Shu. 2023. Can LLM-Generated Misinformation Be Detected?. In *Proc. of International Conference on Learning Representations*.

[8] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234* (2022).

[9] Yaqian Dun, Kefei Tu, Chen Chen, Chunyan Hou, and Xiaojie Yuan. 2021. Kan: Knowledge-aware attention network for fake news detection. In *Proc. of AAAI conference on artificial intelligence*, Vol. 35. 81–89.

[10] Thomas Felber. 2021. Constraint 2021: Machine learning models for COVID-19 fake news detection shared task. *arXiv preprint arXiv:2101.03717* (2021).

[11] Beizhe Hu, Qiang Sheng, Juan Cao, Yuhui Shi, Yang Li, Danding Wang, and Peng Qi. 2024. Bad actor, good advisor: Exploring the role of large language models in fake news detection. In *Proc. of the AAAI Conference on Artificial Intelligence*, Vol. 38. 22105–22113.

[12] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: Robust AI-Text Detection via Adversarial Learning. In *Advances in Neural Information Processing Systems*.

[13] Camille Koenders, Johannes Filla, Nicolai Schneider, and Vinicius Woloszyn. 2021. How vulnerable are automatic fake news detection methods to adversarial attacks? *arXiv preprint arXiv:2107.07970* (2021).

[14] Sejong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent Features of Rumor Propagation in Online Social Media. In *IEEE International Conference on Data Mining*.

[15] Thai Le, Suhang Wang, and Dongwon Lee. 2020. Malcom: Generating malicious comments to attack neural fake news detection models. In *Proc. of IEEE International Conference on Data Mining*. IEEE, 282–291.

[16] Hui Liu, Wenya Wang, Haoru Li, and Haoliang Li. 2024. Teller: A trustworthy framework for explainable, generalizable and controllable fake news detection. In *Proc. of Association for Computational Linguistics*.

[17] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *International Joint Conference on Artificial Intelligence*.

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proc. of International Conference on Learning Representations*.

[19] Ahmadreza Mosallanezhad, Mansooreh Karami, Kai Shu, Michelle V Mancenido, and Huan Liu. 2022. Domain adaptive fake news detection via reinforcement learning. In *Proc. of ACM Web Conference*. 3632–3640.

[20] Qiong Nan, Danding Wang, Yongchun Zhu, Qiang Sheng, Yuhui Shi, Juan Cao, and Jintao Li. 2022. Improving Fake News Detection of Influential Domain via Domain-and Instance-Level Transfer. In *Proc. of International Conference on Computational Linguistics*. 2834–2848.

[21] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic Detection of Fake News. In *Proc. of International Conference on Computational Linguistics*. 3391–3401.

[22] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. In *Proc. of Association for Computational Linguistics*. 231–240.

[23] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proc. of Empirical Methods in Natural Language Processing*. 2931–2937.

[24] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530* (2024).

[25] Julio CS Reis, André Correia, Fabrício Murai, Adriano Veloso, and Fabrício Benevenuto. 2019. Supervised learning for fake news detection. *IEEE Intelligent Systems* 34, 2 (2019), 76–81.

[26] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. defend: Explainable fake news detection. In *Proc. of ACM SIGKDD international conference on knowledge discovery & data mining*. 395–405.

[27] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data* 8, 3 (2020), 171–188.

[28] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* (2018).

[29] Herun Wan, Shangbin Feng, Zhaoxuan Tan, Heng Wang, Yulia Tsvetkov, and Minnan Luo. 2024. Dell: Generating reactions and explanations for llm-based misinformation detection. In *Proc. of Association for Computational Linguistics*.

[30] Haoran Wang, Yingtong Dou, Canyu Chen, Lichao Sun, Philip S Yu, and Kai Shu. 2023. Attacking fake news detectors via manipulating news social engagement. In *Proc. of the ACM Web Conference*. 3978–3986.

[31] Jiaying Wu and Bryan Hooi. 2023. Fake News in Sheep's Clothing: Robust Fake News Detection Against LLM-Empowered Style Attacks. *arXiv preprint arXiv:2310.10830* (2023).

[32] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*.

[33] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large Language Models as Optimizers. In *Proc. of International Conference on Learning Representations*.

[34] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large Language Models are Human-Level Prompt Engineers. In *Proc. of International Conference on Learning Representations*.

[35] Zhixuan Zhou, Huankang Guan, Meghana Moorthy Bhat, and Justin Hsu. 2019. Fake news detection via NLP is vulnerable to adversarial attacks. *arXiv preprint arXiv:1901.09657* (2019).

[36] Yongchun Zhu, Qiang Sheng, Juan Cao, Shuokai Li, Danding Wang, and Fuzhen Zhuang. 2022. Generalizing to the future: Mitigating entity bias in fake news detection. In *Proc. of ACM SIGIR Conference on Research and Development in Information Retrieval*. 2120–2125.

A APPENDIX

A.1 Baseline Implementation

For consistency, experiments replicating existing baselines maintained fixed settings for learning, backbone network, and other relevant parameters. Augmented variations for UDA were generated using back-translation through German. RADAR employed adversarial training to learn paraphrases, utilizing the maximization of the binary cross-entropy loss of a fake news detector as a reward signal, with the T5-large model serving as the paraphraser. Named entities for ENDEF were extracted using the bert-base-NER model from Hugging Face. For SheepDog, augmentations were generated following the original paper's methodology, using the prompt format illustrated in Figure 10 and four tones: "objective and professional," "neutral," "emotionally triggering," and "sensational".

> Rewrite the following article in a/an [tone]: [news article]

Figure 10: Prompt for style-conversion. The "tone" part will be filled with the desired tone, and the "news article" part will contain the original news text.

A.2 Examples of Style-Conversion Prompt

Figure 11 showcases examples generated by the selected style-conversion prompt, demonstrating how fake news content can be modified while preserving its meaning. These examples demonstrate the impact of style transformation on fake news detection.

A.3 Prompt Examples of Other Attack Scenarios

We have conducted comparison experiments with more diverse attack scenarios: (1) Adversarial prompt: Given a news article and its label, the prompt instructs the LLM to rewrite the article to evade detection as the given label. (2) Summarization prompt: A prompt instructing the LLM to summarize the news article without incorporating stylistic guidance. (3) In-Context prompt: A prompt providing an example of a recent, real CNN article, instructing the LLM to rewrite the given article in the same style. Here we provide the example prompt for each attack prompt as figures. (4) Adversarial Paraphraser: The attack utilizes a paraphraser adversarially trained on the training dataset.

> Original: Updated: ASPEN 2014 Senator John McCain is meeting with his top campaign advisers and donors here...
>
> Rewrite the following article in a sarcastic and mocking tone: Oh, how exciting! Senator John McCain is gracing us with his presence in Aspen, meeting with his elite group of advisers and donors...
>
> Rewrite the following article in a chaotic and disorganized tone: Senator John McCain is like, in Aspen, meeting with his campaign peeps and rich donors, you know?...
>
> Rewrite the following article in a nonsensical and absurdly humorous tone with a touch of surrealism and a sprinkle of silliness : Senator John McCain found himself in a pickle. He was surrounded by his top campaign advisers and donors in the mystical town of Aspen...

Figure 11: Generated example input created using the selected style-conversion prompt.

> <EXAMPLES>: [news article]
> Determine whether the given <EXAMPLES> content is fact or fake, mentioning the style of the article that led to that decision following format:
> Answer: {Fact or Fake}
> Reason: {one-sentence reason}

> Your current article is: [news article]
> But this article is detected as [Answer] due to the following [Reason].
> Based on the above information, rewrite a new improved article not to be detected as [news label], maintaining the original content, as follows:

Figure 12: Example input for Adversarial prompt model. First, use the prompt above to extract the answer and reason for the authenticity of the given news. Next, utilize the prompt below to rephrase the news to evade detection based on the given reason. The "news article" section contains the original news text, and the "news label" part contains the corresponding label.

> Summarize the following article, ensuring the content remains the same: [news article]

Figure 13: Example input for Summarization prompt model. The "news article" part is filled with the original news text.

> <EXAMPLES>: [news article example] Rewrite the following article as the writing style of <EXAMPLES> : [news article]

Figure 14: Example input for In-Context prompt model. The "news article example" section is filled with an example from a specific publisher, while the "news article" section contains the original news text.