Content Fuzzing for Escaping Information Cocoons on Digital Social Media

Yifeng He[1] **and Ziye Tang**[2] **and Hao Chen**[3]
[1]Department of Computer Science, [2]Department of Communication
University of California, Davis
[3]University of Hong Kong
{yfhe, szytang}@ucdavis.edu, chenho@hku.hk

Abstract

Information cocoons on social media limit users' exposure to posts with diverse viewpoints. Modern platforms use stance detection as an important signal in recommendation and ranking pipelines, which can route posts primarily to like-minded audiences and reduce cross-cutting exposure. This restricts the reach of dissenting opinions and hinders constructive discourse. We take the creator's perspective and investigate how content can be revised to reach beyond existing affinity clusters. We present CONTENTFUZZ, a confidence-guided fuzzing framework that rewrites posts while preserving their human-interpreted intent and induces different machine-inferred stance labels. CONTENTFUZZ aims to route posts beyond their original cocoons. Our method guides a large language model (LLM) to generate meaning-preserving rewrites using confidence feedback from stance detection models. Evaluated on four representative stance detection models across three datasets in two languages, CONTENTFUZZ effectively changes machine-classified stance labels, while maintaining semantic integrity with respect to the original content.

1 Introduction

Social media platforms increasingly mediate how people access information. However, selective exposure often confines users to highly homogeneous content environments, known as information cocoons (He et al., 2023b; Zhou, 2025). Such confinement narrows individuals' perspectives and limits its informational diversity. Moreover, information cocoons can hinder intellectual growth, reinforce social segmentation, and contribute to emotional or psychological harm (Simpson and Mazzeo, 2017; Napoli and Dwyer, 2018). These cocoons often arise because social media platforms restrict the range of viewpoints that users encounter. Users naturally gravitate toward content sharing a similar stance on various topics, especially early in their interactions with a platform. While this helps recommender systems learn user preferences and deliver personalized content to increase engagement (He, 2022), it also creates a feedback loop in which users are repeatedly exposed to content aligned with their existing beliefs, deepening the cocoon over time.

Information cocoons affect both users and content creators. For users, confinement to homogeneous information environments limits exposure to diverse viewpoints, reinforcing existing biases and amplifying ideological polarization (Garimella et al., 2018). Cross-cutting exposure supports healthier deliberation and reduces affective polarization, while homogeneous and repetitive content can intensify emotional stress and reduce well-being (Alzeer, 2017). For content creators and publishers, these dynamics impose practical constraints: posts circulate primarily within affinity clusters, making it difficult for high-quality content to reach broader or cross-cutting audiences. Consequently, enabling content to escape information cocoons is important for both improving users' informational diversity and enhancing the visibility of creators' messages.

Despite this importance, escaping information cocoons remains technically challenging. Recommendation pipelines and ranking mechanisms operate as black boxes, making it difficult to determine how subtle, semantics-preserving edits influence a post's exposure. Even minor phrasing changes can shift downstream model behavior (Zhang et al., 2025b; He et al., 2025b), yet existing research largely attempts to mitigate cocoon effects through platform-side algorithmic interventions (Krause et al., 2024; Li et al., 2025a). Ma et al. (2025), for example, analyze how diversity-oriented ranking and re-ranking algorithms affect homogenization dynamics and propose algorithmic adjustments to mitigate these effects.

While such algorithmic interventions provide valuable insights into cocoon formation, they re-

Findings of the Association for Computational Linguistics: ACL 2026, pages 11253–11271

main fundamentally platform-controlled. Individual users and content creators cannot modify recommender algorithms, nor do they have visibility into how their posts are filtered, ranked, or delivered. As a result, users and creators have limited agency to expand content reach beyond existing affinity clusters. This gap motivates content-wise approaches that operate independently of platform algorithms. From the creator's perspective, *escaping information cocoons*, achieved by identifying semantics-preserving rewrites that keep a post's human-interpreted stance but change a stance analyzer's predicted label, offers a practical mechanism for broadening cross-group exposure without relying on opaque platform-side changes.

We introduce ContentFuzz, a novel automated content-wise framework to mitigate information cocoons. Our approach targets stance detection models, which constitute a core signal in social media recommendation pipelines for assessing ideological orientation and structuring public-opinion discourse on contentious topics (Hitlin et al., 2019; Zhang et al., 2024a; Muthusami et al., 2025). We adapt fuzzing, a methodology from software testing, to iteratively discover such rewrites. Inspired by recent advances in LLM jailbreak fuzzing (Yu et al., 2024; Liu et al., 2024), ContentFuzz leverages confidence of the stance analysis as feedback to guide a generative LLM in producing semantic-preserving rewrites. Through this feedback-guided process, ContentFuzz reliably alters machine-classified stance labels while preserving the post's human-interpreted stance, thereby enabling content to reach audiences outside its existing cocoon. ContentFuzz is model-agnostic, cross-lingual, cross-topic, and readily adaptable to a wide range of social media scenarios. In our experiments across three real-world datasets in two major languages and four stance detection models, ContentFuzz consistently enables posts to escape information cocoons with robust semantic integrity and fluency in the generated rewrites. To the best of our knowledge, ContentFuzz is the first content-side computational approach toward mitigating information cocoon effects.

2 Background and related work

2.1 Information cocoons

Information cocoons arise when algorithmic curation and selective exposure confine users to homogeneous content environments, limiting informational diversity and reinforcing existing beliefs (He et al., 2023b; Zhou, 2025). Cocoon effects have been documented in news sharing (Du, 2024), video platforms (Yi, 2023), and social media (Chen et al., 2025; Wang et al., 2025a), with Piao et al. (2023) attributing their emergence to human–AI adaptive dynamics. Current mitigation strategies are platform-controlled, such as diversity-oriented re-ranking (Ma et al., 2025), and are therefore unavailable to content creators. ContentFuzz explores a complementary, creator-side direction: searching for semantics-preserving rewrites that shift a post's machine-classified stance label. Such shifts probe whether content can cross stance-conditioned filtering boundaries without altering its human-interpreted meaning. We present this search as an iterative, feedback-guided process built on techniques from software fuzzing.

2.2 Fuzzing

Fuzzing is the process of dynamically testing software by iteratively generating random inputs (Miller et al., 1990). Modern software testing widely adopts gray-box coverage-guided fuzzing, which leverages code coverage as feedback to direct the input generation process (Böhme et al., 2016, 2017). Some recent work also explores applying fuzzing to augment language models (Zhao et al., 2023b; Huang et al., 2024; He et al., 2025a). At a high level, fuzzing consists of three core components: iterative input generation, feedback-based selection, and seed scheduling (Zeller et al., 2019).

Input generation. Fuzzing begins from one or more seed inputs and iteratively produces variants through mutation. Rather than exhaustively enumerating all possible inputs, fuzzing aims to efficiently discover transformations that induce new or interesting behaviors (Chen and Chen, 2018; Chen et al., 2019). Fuzzers can also generate structured inputs, extending to various software domains (Zhang et al., 2024b; Rong et al., 2025; Tu et al., 2026).

Feedback-guided selection. After generating a new input, the fuzzer executes the target system and observes its behavior. In gray-box settings, this signal only needs to correlate with progress toward a desired outcome (Böhme et al., 2016; Rong et al., 2022). Inputs triggering interesting new behaviors are retained as seeds for future iterations.

Seed scheduling. Given a pool of candidate seeds, fuzzers prioritize which inputs to mutate next based on their historical performance. Seed

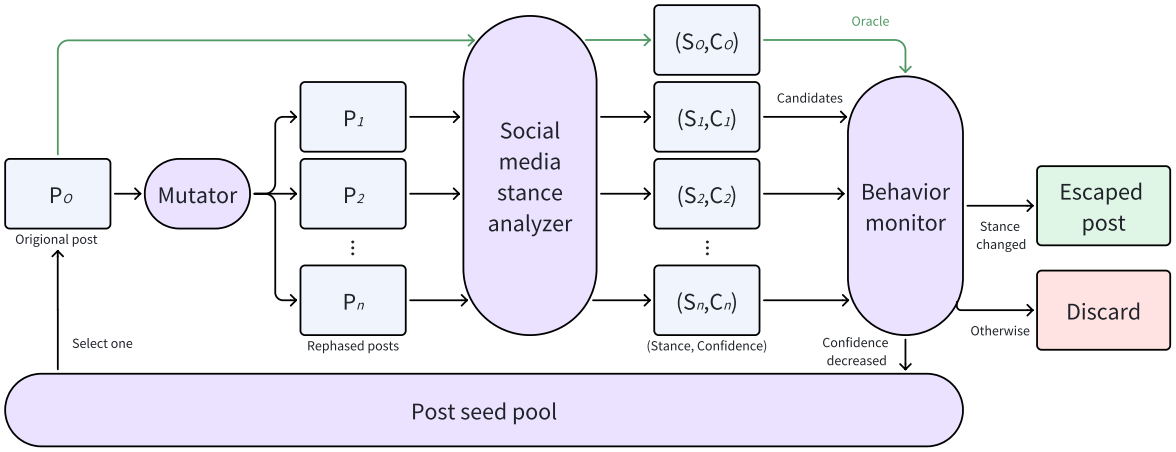

Figure 1: Post content generation in ContentFuzz. *Seed* denotes candidate posts stored for mutation.

scheduling estimates a seed's potential to yield bugs, enabling the fuzzer to focus on promising regions of the input space (Woo et al., 2013; Xu et al., 2024). This prioritization is critical for efficiency when compute hours are limited.

Fuzzing for language models. Fuzzing has also been applied to neural networks, including large language models (LLMs). Bugs in LLMs include jailbreaks, which make the models generate harmful, biased, or toxic content (Perez et al., 2022; Chao et al., 2025). Prior work applies fuzzing-style search to mutate jailbreak prompt templates, using learned classifiers or likelihood-based fitness functions as feedback signals to guide mutation (Yu et al., 2024; Liu et al., 2024). Fuzzing LLMs requires novel designs for input generation, behavior monitoring, and seed scheduling, since LLMs differ significantly from traditional software.

2.3 Stance detection

Stance detection, also referred to as stance analysis, is a natural language classification task that aims to identify the stance or attitude of the author expressed in a piece of text towards a specific target or topic (Mohammad et al., 2016; Zhang et al., 2024a). Stance detection is often used in social media analysis and recommendation, where the platforms expose users to content strongly aligned with their own side with stance-conditioned feed ranking (Garimella et al., 2018; Aldayel and Magdy, 2019; Li et al., 2025a). Modern approaches to stance detection use fine-tuned embedding models (Liu et al., 2021; Conforti et al., 2020; Allaway and McKeown, 2020; Liang et al., 2022; Ding et al., 2025) and generative models (Li et al., 2023; Taranukhin et al., 2024; Gatto et al., 2023; Zhao et al., 2024; Lan et al., 2024). In this work, we focus on post-level stance detection.

2.4 LLM-based text rewriting

LLM-based text rewriting has recently been applied to social media content with objectives ranging from content moderation to engagement optimization. Ziegenbein et al. (2024) use reinforcement learning from machine feedback to rewrite inappropriate argumentation while preserving core claims. Gopalakrishna Pillai et al. (2025) rewrite news tweets to control engagement properties, while Juvino Santos et al. (2025) and Wang et al. (2025b) target polarization reduction and toxic language mitigation, respectively. These methods optimize a social property of the text, whether tone, engagement, or toxicity. ContentFuzz pursues a different objective: rewrites that flip a stance analyzer's predicted label while preserving the post's semantic content. The optimization target is the classifier's decision boundary rather than a social property of the text, which aligns ContentFuzz with adversarial robustness testing for stance models.

3 Design

In this section, we describe the design of ContentFuzz. The workflow is depicted in Figure 1 and detailed in Algorithm 1: starting from a single post as the seed, ContentFuzz mutates it into candidates, runs the stance analyzer to obtain a confidence score for each candidate, keeps confidence-lowering candidates for future mutations, and stops when a candidate changes the predicted stance or when iterations are exhausted. We then detail the three key

Algorithm 1 Confidence-guided content fuzzing

```
function ContentFuzz(post, N)
        ▷ post = original post to apply ContentFuzz
        ▷ N = number of allowed iterations for fuzzing
    Scheduler.Add(post, 1.0)
    for i = 1 to N do
        seed ← Scheduler.Select()
        mutants ←
                Mutator.Rewrite(seed.content)
        n_succ ← 0
        for all m ∈ mutants do
            stance, conf ← Analyze(m)
            if stance ≠ seed.stance then
                return m        ▷ Return successful escape
            if conf < seed.conf then
                Scheduler.Add(m, conf)
                n_succ ← n_succ + 1
        Mutator.UpdateEnergy(n_succ, |mutants|)
    return Nothing                    ▷ No escape found
```

components: feedback guidance (Section 3.1), seed scheduling (Section 3.2), and mutation of the selected seeds (Section 3.3).

3.1 Feedback guidance

Challenge. Stance analyzers based on large language models (LLMs) operate as black boxes and are difficult to interpret (Odena et al., 2019). Black-box testing such systems without feedback is inefficient (Böhme et al., 2016), so defining an effective feedback mechanism to guide gray-box fuzzing for these systems is challenging. Previous work (Xie et al., 2019; Odena et al., 2019) tracks internal neuron activations in deep neural networks as coverage metrics. Park et al. (2023) proposed gradient vector coverage, which leverages gradients obtained by partially differentiating the cross-entropy loss function as feedback. These techniques are impractical in our setting because they require access to internal structures of the target model and do not scale to transformer-based LLMs with hundreds of millions or billions of parameters.

3.1.1 Analysis confidence as feedback

Let us reconsider our fuzzing objective. Unlike previous work that seeks to find *all* inputs that trigger unexpected behaviors, we focus on identifying *one* single variant of a given post that changes the target model's predicted stance. Therefore, we do not aim for completeness in our guidance metric. Instead, we require a metric that reliably indicates whether a mutated candidate is closer to escaping the original stance. To this end, we use the *analysis confidence score* returned by the target stance analyzer as our feedback metric. We describe methodologies to obtain confidence scores from two types of stance analyzers: fine-tuned encoder-based classifiers (*e.g.*, BERT (Devlin et al., 2019)) and generative LLMs (*e.g.*, Gemini-2.5 (Comanici et al., 2025)). In the following text, we use $x = \{x_1, \ldots, x_n\}$ to denote the tokens of the input prompt, which contains the post content, the target topic, and the instruction to generate a stance response if any. We use θ to denote the parameters of the target stance analyzer. Using confidence feedback does not require any instrumentation of the target model.

Classifier stance analyzers. Fine-tuning stance analyzers typically adds a softmax-based classification head on top of a pre-trained masked language model (Sun et al., 2019). The encoder maps the input post to a vector representation, which is fed into a linear classification layer to produce a logit z_i for each label $k \in \{\texttt{Favor}, \texttt{Against}, \texttt{Neutral}\}$. The softmax layer then maps these logits into a probability distribution (Bridle, 1989; Hinton et al., 2015), and $\hat{k}$ is the label with the highest probability:

$$P_\theta(k|x) = \frac{\exp(z_k)}{\sum_j \exp(z_j)}, \hat{k} = \arg\max_k P_\theta(k|x).$$

We use the probability of the predicted stance $\hat{k}$ as the analysis confidence score to guide fuzzing:

$$\text{Conf}_{\text{mlm}}(x, \hat{k}) = P_\theta(\hat{k}|x).$$

Generative stance analyzers. Recent work has investigated using generative LLMs for stance analysis (Lan et al., 2024). These approaches prompt the LLM with the post content and the target topic, asking it to generate a response that indicates the stance. Generative causal language models employ autoregressive decoding, predicting one token at a time based on previously generated tokens and the input prompt (Radford et al., 2019; Brown et al., 2020). Each decoding step is a classification task over the vocabulary. Consequently, each predicted token has an associated probability distribution, often exposed as `logprobs` by the LLM serving API. `logprobs` are the natural logarithms of the model-assigned probabilities for each token in the vocabulary and are often used as a measure of generation confidence to mitigate hallucinations (Xu et al., 2025; Zhang et al., 2025a). Let $y = \{y_1, \ldots, y_m\}$ denote the tokens in the generated stance response. Then the `logprobs` for each generated token y_i are

$$l_i = \log p_\theta(y_i|x, y_{<i}).$$

The joint probability of generating a sequence of tokens is the product of the conditional probabilities of generating each token (Bengio et al., 2000; Radford et al., 2019):

$$p_\theta(y|x) = \prod_{i=1}^{m} p_\theta(y_i|x, y_{<i}).$$

By the basic rules of logarithms, we have

$$L(x, y) = \sum_{i=1}^{m} \log p_\theta(y_i|x, y_{<i}) = \sum_{i=1}^{m} l_i.$$

Our feedback for fuzzing generative stance analyzers is the exponential of the joint `logprobs`:

$$\text{Conf}_{\text{clm}}(x, y) = \exp(L(x, y)).$$

3.2 Seed scheduling

After adding the mutated candidates of interest to the seed pool, ContentFuzz selects the next seed post to fuzz from the pool. Our goal is to identify a mutated candidate that escapes the original stance in as few iterations as possible, so we prioritize seeds that are more likely to lead to successful escapes. These seeds have lower analysis-confidence scores, indicating that they lie closer to the decision boundary of the target stance analyzer. Motivated by this observation, we design our seed-scheduling strategy using a min-heap, where ContentFuzz always selects the seed with the lowest confidence score in the entire pool for mutation. We also present and discuss other seed-scheduling design choices in Section 5.4, where we compare the effectiveness and efficiency of alternative strategies.

3.3 Mutation

3.3.1 LLM-based rewriting

After selecting a seed post, ContentFuzz mutates its content to generate new candidate posts. To achieve this, we design an LLM-based mutator with a strict prompt dedicated solely to rewriting. Unlike software fuzzing with multiple mutators (Fioraldi et al., 2022), we enforce a single rewrite mutation in ContentFuzz to ensure the semantic integrity of the posts. After selecting the seed, we wrap its content in templates (Appendix Figure 5) and send it to an instruction-tuned LLM to produce mutated candidates.

To accelerate exploration and avoid frequent mutation failures, which can terminate fuzzing at an early stage if the pool becomes depleted, we allow the mutator to generate multiple candidates in a single mutation step. In ContentFuzz, we let the mutator generate 5 candidates, and evaluate each against the target stance analyzer individually.

In ContentFuzz, we use Gemini-2.5-Flash-Lite (Comanici et al., 2025). The mutator performs constrained paraphrasing under a strict prompt template (Appendix Figure 5), a narrow task that does not require external knowledge or multi-step reasoning. Recent work validates LLM-based rewriting in closely related settings (Ziegenbein et al., 2024; Gopalakrishna Pillai et al., 2025). To maximize fuzzing throughput and minimize cost, we choose a smaller and faster model to reduce the overhead of using an LLM in the fuzzing process while maintaining competitive performance. To further increase fuzzing throughput, we disable the model's chain-of-thoughts reasoning capability by setting the thinking-token budget to 0.

3.3.2 Temperature scheduling

Temperature in generative LLMs is commonly used to control the level of creativity (Xu et al., 2022; Peeperkorn et al., 2024; Renze and Guven, 2024; Zhu et al., 2024). However, deciding on a fixed temperature for ContentFuzz is challenging. Different social-media platforms and different topics may require different levels of creativity in rewriting. Moreover, ContentFuzz implements only a single, strict mutation operator, for which a fixed temperature may lead to suboptimal exploration-exploitation trade-offs (Böhme et al., 2017; Rong et al., 2022; Luo et al., 2023). To address these challenges, we propose *temperature scheduling*, which dynamically adjusts the temperature during fuzzing.

We discretize the range of temperatures (Google, 2025) into a finite set $\mathcal{T} = \{0.0, 0.1, \ldots, 2.0\}$. For each temperature $t \in \mathcal{T}$, we assign the initial energy value $E_t = 1.0$ for a uniform prior sampling probability. At each fuzzing iteration, we randomly select a temperature t from $\mathcal{T}$ with probability

$$P(t) = \frac{E_t}{\sum_{t' \in \mathcal{T}} E_{t'}}.$$

Suppose the mutator generates N candidates using temperature t, and s of them successfully reduce the analysis confidence compared with their parent seed. We update the energy of t by the mutation success rate of the current iteration

$$E_t \leftarrow E_t + \frac{s}{N}.$$

This adaptive scheduling allows ContentFuzz to dynamically select temperatures that have historically produced higher-quality variations. With temperature scheduling, ContentFuzz seamlessly generalizes across social-media platforms, topics, and target stance analyzers without manual tuning.

4 Experimental setup

4.1 Datasets

We conduct experiments on three stance detection datasets spanning multiple social-media platforms and two languages: SemEval2016-Task6 (Sem16), VAST, and C-STANCE. Sem16 (Mohammad et al., 2016) contains English tweets on six targets. VAST (Allaway and McKeown, 2020) is collected from the Room for Debate section of the *New York Times* and contains English articles on 304 unique targets. C-STANCE (Zhao et al., 2023a) is a Chinese dataset collected from Weibo with 48 126 targets. C-STANCE includes two subtasks: C-STANCE-A for target-based stance detection and C-STANCE-B for domain-based stance detection; we use C-STANCE-A in our experiments. All datasets are expert-annotated, each with a three-class stance scheme that we normalize to a unified set of labels: `Favor`, `Neutral`, and `Against`. Sem16 uses FAVOR/AGAINST/NONE; VAST uses integer labels 0/1/2 (con/pro/neutral); C-STANCE-A uses Chinese labels (支持/反对/中立). All mappings are one-to-one with no granularity reduction.

4.2 Targeted stance analyzers

We evaluate ContentFuzz on three styles of stance analyzers: encoder-based models, zero-shot models, and prompt-engineering models. We select representative models for each style and describe their details in this section. We provide the initial performance of these models in Appendix Table 5.

Encoder. We use BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as our target encoder-based stance analyzers. For English datasets Sem16 and VAST, we use the released `bert-base-uncased` and `roberta-base` checkpoints. For C-STANCE-A, we use the `chinese-bert-wwn` and `chinese-roberta-wwm-ext` checkpoints pre-trained on Chinese corpora by Cui et al. (2020, 2021). Details of these models are provided in Appendix A.1.1.

Zero-shot. We use the term *zero-shot* to refer to generative LLMs used without any parameter or architecture modification (Radford et al., 2019; Zhao et al., 2024), *i.e.*, without fine-tuning, prompt engineering, or in-context learning. We directly prompt an LLM with the instructions in Appendix Figure 4 to perform stance detection. We use Google Gemini-2.5-Flash-Lite (Comanici et al., 2025) as our zero-shot model. We set the temperature to 0 for more deterministic outputs and better reproducibility. We apply guided decoding (Scholak et al., 2021) to restrict the output space to valid stance only.

Prompt engineering. Recent work also explores stance detection using tailored prompts (Li et al., 2023; Lan et al., 2024; Taranukhin et al., 2024). We evaluate our approach on COLA (Lan et al., 2024), the current state-of-the-art prompt-engineering method. In COLA, LLMs are assigned distinct roles that form a collaborative system for analyzing stances in a given post. We directly adapt the released open-source implementation of COLA. We use Gemini-2.5-Flash-Lite for COLA.

4.3 Research questions

To validate the effectiveness of ContentFuzz, we design experiments to answer the following research questions: **RQ1** How effective is ContentFuzz across different stance analyzers and datasets? **RQ2** Do rewrites generated for one analyzer transfer to other unseen ones? **RQ3** How does temperature scheduling impact the effectiveness of ContentFuzz? **RQ4** How does seed scheduling influence the performance of ContentFuzz?

5 Evaluation results

5.1 Performance evaluation

In this section, we evaluate the performance of ContentFuzz across three aspects: success rate, semantic integrity, and fluency. Success rate measures the effectiveness of ContentFuzz in rewriting posts to escape information cocoons, while semantic integrity and fluency assess the quality of the generated rewrites by ContentFuzz.

Escape success rate. We measure the escape success rate (ESR) as the percentage of posts that are classified correctly by the targeted stance analyzer before fuzzing, but are misclassified after being rewritten by ContentFuzz. Let D_{corr} denote the set of correctly classified posts, and let CF denote the ContentFuzz function. Then, for all $p \in D_{\text{corr}}$,

$$ESR = \frac{|\{p | p.stance \neq CF(p).stance\}|}{|D_{\text{corr}}|}.$$

Analyzer	ESR	BERTScore	PPL	PPLr
Sem16				
BERT	0.563	0.889	71.335	0.422
RoBERTa	0.670	0.876	69.420	0.360
Zero-shot	0.773	0.885	112.634	0.601
COLA	0.480	0.882	52.247	0.458
VAST				
BERT	0.883	0.878	10.041	0.322
RoBERTa	0.708	0.869	9.789	0.317
Zero-shot	0.655	0.892	24.448	0.749
COLA	0.410	0.896	16.787	0.537
C-STANCE-A				
BERT	0.910	0.752	21.242	0.164
RoBERTa	0.879	0.774	16.717	0.163
Zero-shot	0.737	0.750	34.016	0.312
COLA	0.750	0.761	49.775	0.376

Table 1: Performance evaluation of ContentFuzz.[1]

BERTScore. BERTScore (Zhang et al., 2020) is a widely used metric to evaluate the semantic similarity between texts. BERTScore uses an encoder model to compute contextual sentence embeddings for both the original post and the rewritten post. We report the mean F1 score over successfully rewritten posts as the semantic integrity score.

Perplexity. Perplexity (PPL) (Jelinek et al., 1977) measures the model's uncertainty in predicting the next token in a sequence, which provides a sense of fluency for generated text. We follow AutoDAN (Liu et al., 2024) to report the perplexity of the rewritten posts. Furthermore, we develop *perplexity ratio* (PPLr) to measure the fluency of the generated rewrites relative to their original posts. Since absolute perplexity is sensitive to topic, style, and language-specific token distributions, directly comparing PPL values across different posts or languages can be misleading. PPLr isolates the fluency change introduced by the rewriting process itself. For each post p that is successfully rewritten,

$$PPLr = \frac{PPL(CF(p))}{PPL(p)}.$$

We report the mean over the central 95% of values to reduce the influence of outliers.

Our evaluation results are summarized in Table 1. ContentFuzz with Gemini-2.5-Flash-Lite is effective across all targeted stance analyzers and datasets, achieving higher ESRs while maintaining strong semantic integrity and low perplexity. Among the targeted stance analyzers, the zero-shot LLM analyzer is the most robust to ContentFuzz. The ESRs on zero-shot analyzers are lower than those of the other analyzers, and the quality of the generated rewrites is also slightly lower. Because the zero-shot analyzer is more robust, successfully escaping posts require more aggressive rewrites that deviate further from typical language patterns, resulting in higher absolute perplexity (*e.g.* 112.634 on Sem16). However, the perplexity ratio (PPLr) remains well below 1.0 (*e.g.* 0.601), confirming that the rewrites are still fluent relative to their originals; the high absolute PPL reflects the short, informal, and topically specific nature of the original Sem16 tweets. We also observe that Chinese posts are easier to rewrite to escape information cocoons than English posts. However, this comes at the cost of slightly lower semantic integrity. We also compare against state-of-the-art adversarial attack methods in Appendix A.4, where we achieve 51% relative improvement in success rate and over 90% relative lower perplexity for generated rewrites. We also provide case studies in Appendix A.5 to illustrate how ContentFuzz iteratively rewrites a post to change the prediction of the targeted analyzer.

NLI-based contradiction analysis. BERTScore measures contextual similarity but cannot detect semantic inversions (*e.g.* negation). Natural language inference (NLI) (Bowman et al., 2015) is the task of determining whether a hypothesis is entailed by, contradicts, or is neutral with respect to a given premise. Following Kambhatla et al. (2024), who use NLI to verify meaning preservation in text rewriting, we verify that rewrites do not contradict the originals, and in reverse. We use `cross-encoder/nli-deberta-v3-large` (He et al., 2023a) for English and `MoritzLaurer/mDeBERTa-v3-base-mnli-xnli` (Laurer et al., 2024) for Chinese. Table 2 summarizes the results. In the forward direction (original entails rewrite), the entailment rate exceeds 80% on every dataset (93.88% in aggregate) while the contradiction rate remains below 2%, providing direct evidence that ContentFuzz rewrites preserve meaning and complementing the embedding-based similarity captured by BERTScore with an explicit logical consistency check.

Finally, we analyze whether fuzzing progress reduces the semantic integrity of the generated rewrites. Figure 2 shows the semantic integrity of successfully rewritten posts over fuzzing iterations, measured by BERTScore F1 on the Sem16

[1]Due to COLA running too slowly to obtain final results, we sampled 100 posts for this evaluation.

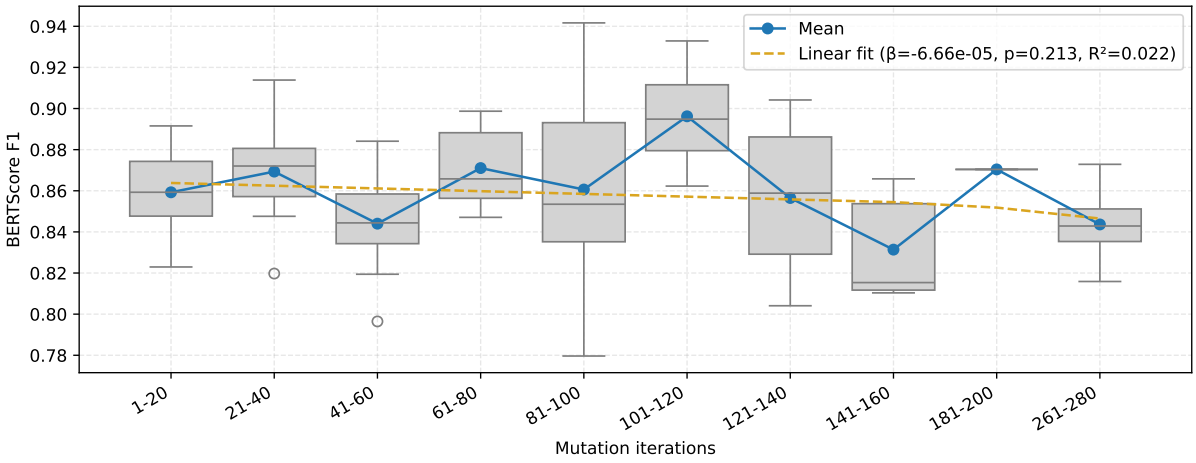

Figure 2: Semantic integrity over fuzzing iterations.

Direction	Dataset	Ent.	Neu.	Con.
→	Sem16	80.62	17.50	1.88
	VAST	97.56	2.28	0.16
	C-STANCE	94.03	5.38	0.59
	All	**93.88**	**5.54**	**0.59**
←	Sem16	66.67	29.06	4.27
	VAST	92.04	7.42	0.54
	C-STANCE	92.44	6.58	0.99
	All	**90.46**	**8.41**	**1.12**

Table 2: NLI-based contradiction analysis on all successfully rewritten pairs. The forward direction is from original to rewrite. Ent.: Entailment, Neu.: Neutral, Con.: Contradiction. Values are in %.

dataset with the fine-tuned RoBERTa. We observe that the BERTScore remains relatively stable as the number of fuzzing iterations increases. To quantify this trend, we fit a linear regression between mean BERTScore and iteration index and find only a negligible negative coefficient ($\beta = -6.66 \times 10^{-5}$), which is not statistically significant ($p = 0.213$) and explains little variance ($R^2 = 0.022$). Indicated by the results, we cannot conclude that more fuzzing iterations lead to systematic semantic degradation.

5.2 Cross-model success rate

We investigate whether rewrites found for one targeted stance analyzer transfer to other unseen analyzers. To this end, we measure the extent to which escaping posts produced by fuzzing against one targeted stance analyzer can also successfully escape other unseen stance analyzers (Papernot et al., 2016; Liu et al., 2024). For cross-model transferability, we take rewrites produced by fuzzing one target model and evaluate the misclassification rate on other unseen models, defined as $1 - \text{Acc}$, where Acc is accuracy on those unseen models.

We present the cross-model transferability results in Figure 3. We observe that models sharing the same architecture exhibit higher cross-model transferability. For example, the fine-tuned BERT and RoBERTa demonstrate higher transferability with each other. Furthermore, we find that COLA's cross-model success rate is very low for the Sem16 dataset, but relatively high for the VAST and C-STANCE-A datasets. We attribute this discrepancy to the fact that COLA uses manually designed expert roles for collaborative debates around the six topics in Sem16. However, its performance and robustness do not generalize well to datasets with different topics and writing styles. In addition, zero-shot LLM-based analyzers exhibit lower cross-model transferability than fine-tuned encoder-based models, indicating stronger robustness of LLMs against semantic-preserving rewrites. Cross-model transferability is higher between architecturally similar models (*e.g.* BERT ↔ RoBERTa), while zero-shot LLM-based analyzers exhibit stronger robustness against transferred rewrites.

5.3 Effects of temperature scheduling

To analyze the effects of temperature scheduling, we fix the seed scheduling strategy to *priority queue* and fuzz the target stance analyzer with and without temperature scheduling. For fuzzing without temperature scheduling, we set the temperature to a constant value of 1.0, which is the default value when accessing LLM APIs. We report the mean, median, and standard deviation (std) of iterations required for successful posts. We use RoBERTa as the targeted stance analyzer on the Sem16 dataset, and the

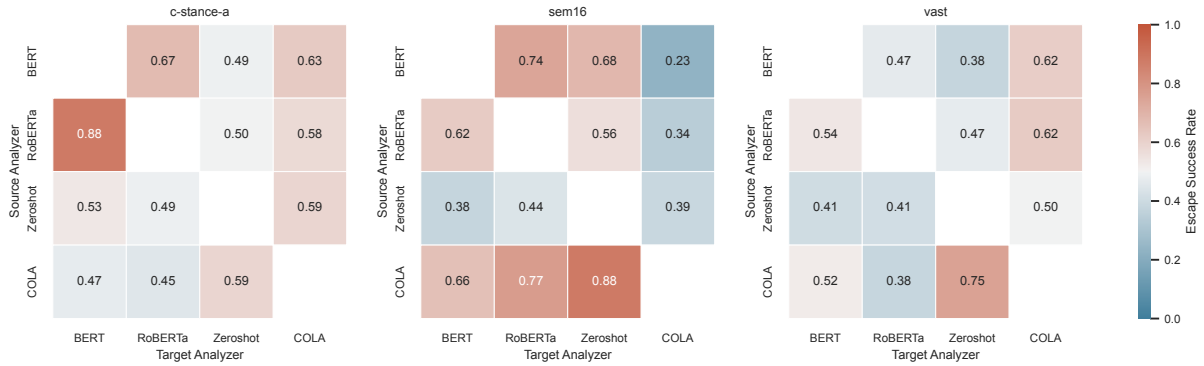

Figure 3: Cross-model transferability.

Temperature	ESR	mean	median	std
1.0	0.620	16.702	2	36.362
Scheduling	0.670	15.324	2	36.931

Table 3: Effects of temperature scheduling.

Scheduling	ESR	mean	median	std
FIFO	0.620	22.798	3	47.830
Random	0.645	18.326	3	35.513
Weighted	0.665	15.985	2	38.140
Priority	0.670	15.324	2	36.931

Table 4: Effects of seed scheduling.

detailed settings are provided in Appendix A.3.1.

The statistics of the experiments are summarized in Table 3. We observe a clear advantage of temperature scheduling over using a constant temperature of 1.0. Since different posts have different wording, sentiments, styles, and stances, fixing the temperature to a constant value limits the diversity of generated content, and thus requires more fuzzing iterations to successfully rewrite the posts. In contrast, letting the fuzzer adapt the temperature during the fuzzing process allows it to generate more diverse content, which improves the efficiency and effectiveness of ContentFuzz.

5.4 Effects of seed scheduling

Modern fuzzers (Fioraldi et al., 2020, 2022) support multiple seed scheduling strategies, and users can choose different strategies based on their needs and their application domain. To this end, we also designed multiple seed scheduling strategies for ContentFuzz to accommodate different post topics on social media. We implemented and evaluated four different seed scheduling strategies: FIFO, random, weighted, and priority scheduling. The priority scheduling strategy is described in Section 3.2, and the others are detailed in Appendix A.3.2. We follow the same settings as in Section 5.3.

From Table 4, we observe that the priority scheduling strategy outperforms all other strategies in ESR. However, weighted probability scheduling is more efficient in terms of the maximum number of iterations required for successful posts, with the lowest standard deviation as well. This indicates that different seed scheduling strategies have different advantages. We select the priority scheduling strategy as the default seed scheduling strategy for ContentFuzz, since it achieves the highest ESR.

6 Conclusion

We present ContentFuzz, the first content-focused methodology that enables content creators to mitigate information cocoons on social media platforms. ContentFuzz adopts a gray-box approach that leverages confidence scores from stance analyzers to guide an iterative rewriting process, in which a generative LLM modifies post content. The generated posts preserve the original, human-interpreted stance toward a given social topic, while being classified differently by stance analyzers deployed on social media platforms. ContentFuzz effectively generates diverse rewrites that escape information cocoons with high success rates, while maintaining the original semantics of the posts. We believe ContentFuzz represents a promising new direction in responsible AI for social media research, with a particular focus on mitigating information cocoons. Our source code is available at `https://github.com/EYH0602/ContentFuzz`.

Limitations

While ContentFuzz demonstrates promising results in mitigating information cocoons on social media platforms, several limitations warrant consideration. First, the current design of ContentFuzz focuses exclusively on stance detection, which represents only one of the predictive components in modern recommender systems. Future work could extend the methodology to additional predictors or to end-to-end recommender systems. Second, we do not extensively optimize ContentFuzz or tune its hyperparameters to maximize success rates, beyond designing temperature and seed scheduling strategies. As ContentFuzz is the first work exploring content rewriting for escaping information cocoons, our primary goal is to demonstrate feasibility rather than achieve optimal performance. Third, ContentFuzz relies on confidence scores (`logprobs`) from stance detection models as feedback to guide the rewriting process. However, for some newer proprietary LLMs, these `logprobs` are not directly accessible. Fun-tuning (Labunets et al., 2025) proposes estimating `logprobs` using fine-tuning loss with a very small learning rate, which could serve as an alternative.

Fourth, our evaluation relies on computational metrics rather than direct human annotation of meaning preservation. While these metrics provide strong and complementary evidence, they do not constitute a direct measurement of whether humans perceive the stance and intent as unchanged. Nevertheless, our small-scale human evaluation (Appendix A.5) confirms that the rewrites preserve the original meaning. Finally, our evaluation is limited to empirical studies on public datasets and the aforementioned computational analysis metrics. We do not examine downstream real-world impacts of deploying posts produced by ContentFuzz on production social media platforms due to limited platform accessibility.

Ethical considerations

This work investigates how content creators may automatically rewrite posts to change the prediction of automated stance analyzers, and thereby escape algorithmically induced information cocoons. The objective is to analyze and expose structural biases in stance-based recommender and moderation pipelines, rather than to facilitate deception, misinformation, or malicious manipulation. Accordingly, the rewriting process is strictly constrained to semantically preserving LLM-based rewrites that maintain the original intent and factual content of the post. ContentFuzz does not generate new content or introduce new claims; it rephrases existing content to probe the limitations of stance-based filtering mechanisms. All experiments are conducted on public datasets and models, without targeting real users, platforms, or deployed production systems. The case study examples are drawn and rewritten verbatim from publicly available research datasets (Mohammad et al., 2016; Allaway and McKeown, 2020; Zhao et al., 2023a); user handles appearing in the original data have been anonymized to prevent identification. These examples cover socially sensitive topics (e.g., abortion, feminism, religion) and are presented solely for research illustration; their inclusion does not reflect the views of the authors. While such techniques might be misused to evade automated moderation, we frame our contribution as a diagnostic and exploratory study intended to improve transparency and robustness in stance-aware recommender systems.

Acknowledgments

This material is based upon work supported by UC Noyce Initiative.

References

Abeer Aldayel and Walid Magdy. 2019. Your stance is exposed! analysing possible factors for stance detection on social media. *Proceedings of the ACM on Human-Computer Interaction*, 3:1 – 20.

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.

Gergana Alzeer. 2017. Cocoons as a space of their own: a case of emirati women learners. *Gender, Place & Culture*, 24(7):1031–1050.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Marcel Böhme, Van-Thuan Pham, Manh-Dung Nguyen, and Abhik Roychoudhury. 2017. Directed greybox fuzzing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 2329 – 2344, New York, NY, USA. Association for Computing Machinery.

Marcel Böhme, Van-Thuan Pham, and Abhik Roychoudhury. 2016. Coverage-based greybox fuzzing as markov chain. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 1032 – 1043, New York, NY, USA. Association for Computing Machinery.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

John Bridle. 1989. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2025. Jailbreaking Black Box Large Language Models in Twenty Queries . In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42, Los Alamitos, CA, USA. IEEE Computer Society.

Peng Chen and Hao Chen. 2018. Angora: Efficient fuzzing by principled search. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 711–725.

Peng Chen, Jianzhong Liu, and Hao Chen. 2019. Matryoshka: Fuzzing deeply nested branches. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 499 – 513, New York, NY, USA. Association for Computing Machinery.

Sihua Chen, Han Qiu, and Wei He. 2025. The information cocoon paradox: fostering unity or fueling divergence? *Humanities and Social Sciences Communications*, 12:859.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

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.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514.

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.

Yuzhe Ding, Kang He, Bobo Li, Li Zheng, Haijun He, Fei Li, Chong Teng, and Donghong Ji. 2025. Zero-shot conversational stance detection: Dataset and approaches. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3221–3235, Vienna, Austria. Association for Computational Linguistics.

Roselyn Du. 2024. News recommendation and information cocoons: The impact of algorithms on news consumption. In *Handbook of Applied Journalism: Theory and Practice*, pages 43–61. Springer Nature Switzerland, Cham.

Andrea Fioraldi, Dominik Maier, Heiko Eißfeldt, and Marc Heuse. 2020. AFL++ : Combining incremental steps of fuzzing research. In *14th USENIX Workshop on Offensive Technologies (WOOT 20)*. USENIX Association.

Andrea Fioraldi, Dominik Christian Maier, Dongjia Zhang, and Davide Balzarotti. 2022. Libafl: A framework to build modular and reusable fuzzers. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 1051 – 1065, New York, NY, USA. Association for Computing Machinery.

Chongyang Gao, Kang Gu, Soroush Vosoughi, and Shagufta Mehnaz. 2024. Semantic-preserving adversarial example attack against BERT. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 202–207, Mexico City, Mexico. Association for Computational Linguistics.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 913 – 922, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Joseph Gatto, Omar Sharif, and Sarah M. Preum. 2023. Chain-of-thought embeddings for stance detection on social media. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4154–4161, Singapore. Association for Computational Linguistics.

Google. 2025. Gemini 2.5 flash-lite. Google Cloud Documentation.

Reshmi Gopalakrishna Pillai, Antske Fokkens, and Wouter van Atteveldt. 2025. Engagement-driven persona prompting for rewriting news tweets. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8612–8622, Abu Dhabi, UAE. Association for Computational Linguistics.

Yiwen Guo, Qizhang Li, and Hao Chen. 2020. Back-propagating linearly improves transferability of adversarial examples. In *Advances in Neural Information Processing Systems*, volume 33, pages 85–95. Curran Associates, Inc.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023a. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Yifeng He. 2022. Big data and deep learning techniques applied in intelligent recommender systems. In *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, pages 1119–1124.

Yifeng He, Jicheng Wang, Yuyang Rong, and Hao Chen. 2025a. FuzzAug: Data augmentation by coverage-guided fuzzing for neural test generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15642–15655, Suzhou, China. Association for Computational Linguistics.

Yifeng He, Luning Yang, Christopher Castro Gaw Gonzalo, and Hao Chen. 2025b. Evaluating program semantics reasoning with type inference in system f. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yiqing He, Darong Liu, Ruitong Guo, and Siping Guo. 2023b. Information cocoons on short video platforms and its influence on depression among the elderly: A moderated mediation model. *Psychology Research and Behavior Management*, 16:2469–2480.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Paul Hitlin, Lee Rainie, and Kenneth Olmstead. 2019. Facebook algorithms and personal data. *Pew Research Center*.

Jiabo Huang, Jianyu Zhao, Yuyang Rong, Yiwen Guo, Yifeng He, and Hao Chen. 2024. Code representation pre-training with complements from program executions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track (EMNLP)*, pages 267–278, Miami, Florida, US. Association for Computational Linguistics.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Yunhan Jia, Yantao Lu, Junjie Shen, Qi Alfred Chen, Hao Chen, Zhenyu Zhong, and Tao Wei. 2020. Fooling detection alone is not enough: Adversarial attack against multiple object tracking. In *International Conference on Learning Representations*.

Lucas Raniére Juvino Santos, Leandro Balby Marinho, Claudio Elizio Calazans Campelo, Filippo Menczer, and Alessandro Flammini. 2025. Can large language models effectively mitigate polarization in social media text? In *Proceedings of the 17th ACM Web Science Conference 2025*, Websci '25, page 348 – 357, New York, NY, USA. Association for Computing Machinery.

Gauri Kambhatla, Matthew Lease, and Ashwin Rajadesingan. 2024. Promoting constructive deliberation: Reframing for receptiveness. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5110–5132, Miami, Florida, USA. Association for Computational Linguistics.

Thorsten Krause, Alina Deriyeva, Jan H. Beinke, Gerrit Y. Bartels, and Oliver Thomas. 2024. Mitigating exposure bias in recommender systems—a comparative analysis of discrete choice models. *ACM Trans. Recomm. Syst.*, 3(2).

Andrey Labunets, Nishit V. Pandya, Ashish Hooda, Xiaohan Fu, and Earlence Fernandes. 2025. Fun-tuning: Characterizing the vulnerability of proprietary llms to optimization-based prompt injection attacks via the fine-tuning interface. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 411–429.

Xiaochong Lan, Chen Gao, Depeng Jin, and Yong Li. 2024. Stance detection with collaborative role-infused llm-based agents. *Proceedings of the International AAAI Conference on Web and Social Media*, 18(1):891–903.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2024. Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli. *Political Analysis*, 32(1):84 – 100.

Ang Li, Bin Liang, Jingqian Zhao, Bowen Zhang, Min Yang, and Ruifeng Xu. 2023. 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.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020a. 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.

Nan Li, Bo Kang, and Tijl De Bie. 2025a. Content-agnostic moderation for stance-neutral recommendations. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, page 1613 – 1623, New York, NY, USA. Association for Computing Machinery.

Qizhang Li, Yiwen Guo, and Hao Chen. 2020b. Practical no-box adversarial attacks against dnns. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Qizhang Li, Yiwen Guo, Xiaochen Yang, Wangmeng Zuo, and Hao Chen. 2025b. Improving transferability of adversarial examples via bayesian attacks. *IEEE Transactions on Circuits and Systems for Video Technology*.

Bin Liang, Zixiao Chen, Lin Gui, Yulan He, Min Yang, and Ruifeng Xu. 2022. Zero-shot stance detection via contrastive learning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2738 – 2747, New York, NY, USA. Association for Computing Machinery.

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*, pages 3152–3157, Online. Association for Computational Linguistics.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Changhua Luo, Wei Meng, and Penghui Li. 2023. Selectfuzz: Efficient directed fuzzing with selective path exploration. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2693–2707.

Qiyao Ma, Menglin Yang, Mingxuan Ju, Tong Zhao, Neil Shah, and Rex Ying. 2025. Breaking information cocoons: A hyperbolic graph-llm framework for exploration and exploitation in recommender systems. *Preprint*, arXiv:2411.13865.

Dongyu Meng and Hao Chen. 2017. MagNet: a two-pronged defense against adversarial examples. In *ACM Conference on Computer and Communications Security (CCS)*, Dallas, TX.

Barton P. Miller, Lars Fredriksen, and Bryan So. 1990. An empirical study of the reliability of unix utilities. *Commun. ACM*, 33(12):32 – 44.

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.

Rathinasamy Muthusami, Kandhasamy Saritha, Kolli Srinivasa Rao, Palanisamy Sugapriya, and G. Saveetha. 2025. Interpretable stance detection in social media via topic-guided transformers. *Discover Artificial Intelligence*, 5:355.

Philip M. Napoli and Deborah L. Dwyer. 2018. U.s. media policy in a time of political polarization and technological evolution. 63.

Augustus Odena, Catherine Olsson, David Andersen, and Ian Goodfellow. 2019. TensorFuzz: Debugging neural networks with coverage-guided fuzzing. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4901–4911. PMLR.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *Preprint*, arXiv:1605.07277.

Leo Hyun Park, Soochang Chung, Jaeuk Kim, and Taekyoung Kwon. 2023. Gradfuzz: Fuzzing deep neural networks with gradient vector coverage for adversarial examples. *Neurocomputing*, 522:165–180.

Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. 2024. Is temperature the creativity parameter of large language models? In *ICCC*, pages 226–235.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinghua Piao, Jiazhen Liu, Fang Zhang, Jun Su, and Yong Li. 2023. Human–ai adaptive dynamics drives the emergence of information cocoons. *Nature Machine Intelligence*, 5(11):1214–1224.

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

Matthew Renze and Erhan Guven. 2024. The effect of sampling temperature on problem solving in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7346–7356, Miami, Florida, USA. Association for Computational Linguistics.

Yuyang Rong, Zhanghan Yu, Zhenkai Weng, Stephen Neuendorffer, and Hao Chen. 2025. *IRFuzzer: Specialized Fuzzing for LLVM Backend Code Generation*, page 1986 – 1998. IEEE Press.

Yuyang Rong, Chibin Zhang, Jianzhong Liu, and Hao Chen. 2022. Valkyrie: Improving fuzzing performance through deterministic techniques. In *2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS)*, pages 628–639.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Courtney C. Simpson and Suzanne E. Mazzeo. 2017. Skinny is not enough: A content analysis of fitspiration on pinterest. *Health Communication*, 32(5):560–567. PMID: 27326747.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18 – 20, 2019, Proceedings*, page 194 – 206, Berlin, Heidelberg. Springer-Verlag.

Maksym Taranukhin, Vered Shwartz, and Evangelos Milios. 2024. Stance reasoner: Zero-shot stance detection on social media with explicit reasoning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15257–15272, Torino, Italia. ELRA and ICCL.

Haoxin Tu, Seongmin Lee, Yuxian Li, Peng Chen, Lingxiao Jiang, and Marcel Böhme. 2026. Cottontail: Large language model-driven concolic execution for highly structured test input generation. In *Proceedings of the 47th IEEE Symposium on Security and Privacy*, SP'26.

Lin Wang, Molin Yang, Alex Wang, Jiayin Zhang, and Sean Xin Xu. 2025a. Understanding the dynamics of information cocoons on social media platforms. In *Proceedings of the Pacific Asia Conference on Information Systems (PACIS 2025)*, Kuala Lumpur, Malaysia. Association for Information Systems (AIS). Completed Research Paper, Paper 19.

Xintong Wang, Yixiao Liu, Jingheng Pan, Liang Ding, Longyue Wang, and Chris Biemann. 2025b. Chinese toxic language mitigation via sentiment polarity consistent rewrites. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35695–35711, Suzhou, China. Association for Computational Linguistics.

Maverick Woo, Sang Kil Cha, Samantha Gottlieb, and David Brumley. 2013. Scheduling black-box mutational fuzzing. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, page 511 – 522, New York, NY, USA. Association for Computing Machinery.

Xiaofei Xie, Lei Ma, Felix Juefei-Xu, Minhui Xue, Hongxu Chen, Yang Liu, Jianjun Zhao, Bo Li, Jianxiong Yin, and Simon See. 2019. Deephunter: a coverage-guided fuzz testing framework for deep neural networks. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2019, page 146 – 157, New York, NY, USA. Association for Computing Machinery.

Frank F. Xu, Uri Alon, Graham Neubig, and Vincent Josua Hellendoorn. 2022. A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, MAPS 2022, page 1 – 10, New York, NY, USA. Association for Computing Machinery.

Hang Xu, Liheng Chen, Shuitao Gan, Chao Zhang, Zheming Li, Jiangan Ji, Baojian Chen, and Fan Hu. 2024. Graphuzz: Data-driven seed scheduling for coverage-guided greybox fuzzing. *ACM Trans. Softw. Eng. Methodol.*, 33(7).

Mengyao Xu, Qiaoyin Gan, Zhenyu Zhu, and Haojun Qin. 2025. Logprobs know uncertainty: Fighting llm hallucinations. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, FSE Companion '25, page 1242 – 1243, New York, NY, USA. Association for Computing Machinery.

Wang Yi. 2023. An analysis of the information cocoon effect of news clients: Today's headlines as an example. *The Frontiers of Society, Science and Technology*, 5(9).

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2024. LLM-Fuzzer: Scaling assessment of large language model jailbreaks. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 4657–4674, Philadelphia, PA. USENIX Association.

Andreas Zeller, Rahul Gopinath, Marcel Böhme, Gordon Fraser, and Christian Holler. 2019. The Fuzzing Book.

Bowen Zhang, Genan Dai, Fuqiang Niu, Nan Yin, Xiaomao Fan, Senzhang Wang, Xiaochun Cao, and Hu Huang. 2024a. A survey of stance detection on social media: New directions and perspectives. *Preprint*, arXiv:2409.15690.

Hongxiang Zhang, Hao Chen, Muhao Chen, and Tianyi Zhang. 2025a. Active layer-contrastive decoding reduces hallucination in large language model generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3046, Suzhou, China. Association for Computational Linguistics.

Hongxiang Zhang, Yifeng He, and Hao Chen. 2025b. Steerdiff: Steering towards safe text-to-image diffusion models. *Preprint*, arXiv:2410.02710.

Hongxiang Zhang, Yuyang Rong, Yifeng He, and Hao Chen. 2024b. Llamafuzz: Large language model enhanced greybox fuzzing. *Preprint*, arXiv:2406.07714.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Chenye Zhao, Yingjie Li, and Cornelia Caragea. 2023a. C-STANCE: A large dataset for Chinese zero-shot stance detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13369–13385, Toronto, Canada. Association for Computational Linguistics.

Chenye Zhao, Yingjie Li, Cornelia Caragea, and Yue Zhang. 2024. ZeroStance: Leveraging ChatGPT for open-domain stance detection via dataset generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13390–13405, Bangkok, Thailand. Association for Computational Linguistics.

Jianyu Zhao, Yuyang Rong, Yiwen Guo, Yifeng He, and Hao Chen. 2023b. Understanding programs by exploiting (fuzzing) test cases. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10667–10679, Toronto, Canada. Association for Computational Linguistics.

Sirui Zhou. 2025. The impact of the information cocoon effect of social media on individuals and societal development. *Interdisciplinary Humanities and Communication Studies*, 1.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy. Association for Computational Linguistics.

Yuqi Zhu, Jia Li, Ge Li, YunFei Zhao, Jia Li, Zhi Jin, and Hong Mei. 2024. Hot or cold? adaptive temperature sampling for code generation with large language models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Timon Ziegenbein, Gabriella Skitalinskaya, Alireza Bayat Makou, and Henning Wachsmuth. 2024. LLM-based rewriting of inappropriate argumentation using reinforcement learning from machine feedback. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4455–4476, Bangkok, Thailand. Association for Computational Linguistics.

A Appendix

A.1 Stance analyzer details

A.1.1 Fine-tuning

We follow the provided split of each dataset for training, validation, and test. For Sem16 and VAST, we fine-tune the models with a learning rate of 2×10^{-5}. For C-STANCE-A, we fine-tune the models with a learning rate of 5×10^{-6}, following Zhao et al. (2023a). We fine-tune all models for 5 epochs with a batch size of 32 and select the checkpoint with the best validation macro-F1. We conduct all fine-tuning on a single NVIDIA A100 40GB GPU. For inference during fuzzing, we use an NVIDIA GeForce RTX 3060 with 12GB memory.

A.1.2 Prompts for stance analysis

System Instruction

```
You are a precise stance classifier. Decide
whether the author's attitude is Favor / Against /
Neutral towards the target target. Be conservative:
if unclear, choose Neutral. ONLY output one word
chosen from Favor, Against, Neutral.
```

Figure 4: The system instruction for generative stance analysis (zero-shot and prompt-engineering).

Model	**C-STANCE-A**		**Sem16**		**VAST**	
	Acc	*F1*	*Acc*	*F1*	*Acc*	*F1*
BERT	0.76	0.76	0.62	0.53	0.70	0.70
RoBERTa	0.78	0.78	0.65	0.62	0.74	0.73
Zero-shot	0.52	0.52	0.58	0.56	0.57	0.56
COLA	0.49	0.41	0.67	0.29	0.41	0.31

Table 5: Performance of different stance analyzers.

A.2 Prompts for content mutation

System Instruction

```
You are a helpful writing assistant and an avid
social media user. Your role is to help the
content creator refine their post to make it
more engaging and shareable. Improve the writing
and flow while **keeping the post's original
meaning intact**. **Do not change the author's
stance** (their position or opinion on the topic)
**or the target topic** of the post. Make sure
to **preserve the original tone, style, and
sentiment** of the writing, maintaining the
author's voice. Only **make minimal edits**:
the goal is to polish the text, not to overhaul
it. Output **only** the revised text, and do
not include any explanations. Always **keep the
content in the same language** as the original
post (no translation or dialect change). Do not
extensively use emojis or hashtags unless they
were present in the original text.
```

Prompt Template

```
The current text is {stance} towards the {target}.
Without changing its meaning, please rewrite the
following text:
```
{text}
```
```

Figure 5: The system instruction and prompt template for LLM-based rewrite mutation.

System Instruction

```
你是一个乐于助人的写作助理，同时也是一个活跃的社交媒体用
户。你的角色是帮助内容创作作者润色他们的帖子，让他们的帖
子变得更加有吸引力和传播性。请在保持帖子本身原意不变的前
提下，提高写作和文章流畅度。请不要改变作者的原本立场（他
们对主题的态度或观点）或帖子的目标主题。请务必保留原有的
语气、风格和观点，保持作者个人表达。只进行最小幅度的修改：
目标是润色文本，而不是重写。输出只提供修改后的文本，不要
附加任何解释。从始至终保持和原文相同的语言（不要翻译或者
转换方言）。除非是原文中已经有的表情符号或话题标签，否则不
要过度使用表情符号或话题标签。
```

Prompt Template

```
当前的文本关于{target}是{stance}的。在不改变其含义的情
况下，请重新写以下文本：
```
{text}
```
```

Figure 6: The Chinese system instruction and prompt template for LLM-based rewrite mutation.

A.3 Experiment settings and details

A.3.1 Temperature scheduling

In this subsection, we evaluate the effectiveness of temperature scheduling and other components of ContentFuzz. Specifically, we analyze their effects from two perspectives:

1. Performance: We follow Section 5.1 to measure the escape success rate (ESR) of ContentFuzz under different configurations.

2. Resource efficiency: We report the distribution of the number of iterations that ContentFuzz needs to rewrite posts.

We perform all ablation studies on 200 tasks randomly sampled from the Sem16 dataset (Mohammad et al., 2016). We fix the maximum number of iterations at 300 for all experiments. We use the fine-tuned RoBERTa (Liu et al., 2019) from Section 4.2 as the targeted stance analyzer.

A.3.2 Seed scheduling

First-in-first-out (FIFO). We implement a simple FIFO queue to store the seed posts as a baseline scheduling strategy, *i.e.*, without scheduling. When the fuzzer considers a seed post interesting, it appends the post to the end of the queue. To select the next seed post to fuzz, the fuzzer takes the seed post at the front of the queue.

Random. The random seed scheduling strategy samples a seed from the seed pool in each iteration. It assigns each seed in the pool equal probability, regardless of their previous fuzzing results.

Weighted. The weighted seed scheduling strategy assigns different weights to different seed posts in the seed pool. We compute the weights from the confidence scores of the targeted stance analyzer on the seed posts. Let s denote a seed post in the seed pool, and let $W(\cdot)$ denote its weight,

$$W(s) = \frac{1}{\text{Conf}(s.content, s.stance)}.$$

Then the probability $P(\cdot)$ of picking the seed s from the seed pool is

$$P(s) = \frac{W(s)}{\sum_{s' \in \text{seed pool}} W(s')}.$$

The strategy samples seeds with lower confidence scores more often, but it can still pick any seed by chance.

A.4 Comparison with adversarial methods

Another line of related work concerns adversarial attacks (Meng and Chen, 2017; Li et al., 2020b; Jia et al., 2020; Guo et al., 2020; Li et al., 2025b). Adversarial attacks on classification models add noise to the original inputs to mislead the model,

producing adversarial examples. Attackers often optimize these perturbations to be imperceptible to humans. For text classification, adversarial examples typically preserve semantics. Although our work does not aim to attack stance analyzers or recommender systems, our content mutation task shares characteristics with adversarial attacks. We compare ContentFuzz with two state-of-the-art adversarial attack methods.

A.4.1 Baselines and experimental settings

BERT-Attack. BERT-Attack (Li et al., 2020a) targets text classification models fine-tuned on BERT. BERT-Attack first identifies vulnerable words in the input text that most influence the model's prediction, then iteratively replaces the ranked words with BERT-based lexical substitutions (Zhou et al., 2019). BERT-Attack aims to minimize the perturbation rate while achieving a high success rate. We use the officially released code[2] in our evaluation.

Reinforce-Attack. Gao et al. (2024) proposed Reinforce-Attack, which generates semantic-preserving adversarial examples against BERT-based classifiers. Reinforce-Attack utilizes a reinforcement learning framework to optimize the generation of adversarial examples, where the attack process is controlled by a reward function rather than heuristic rules. The reward function encourages higher semantic similarity and lower query costs, and the method achieves significantly higher semantic similarity than BERT-Attack while maintaining comparable attack success rates. Because the authors did not release code, we reimplement Reinforce-Attack based on the descriptions in the original paper.

We evaluate these methods on the Sem16, VAST, and C-STANCE-A datasets with a fine-tuned BERT stance analyzer because they do not support other model architectures or languages. To the best of our knowledge, no existing adversarial attack methods can be generalized to encoder-based, zero-shot generative, and prompt-engineering-based stance analyzers as ContentFuzz does.

A.4.2 Result analysis

We present the comparison results in Table 6. ContentFuzz outperforms BERT-Attack and Reinforce-Attack by a large margin in attack success rate (ASR) and fluency (PPL and PPLr) across all three datasets. Although the BERTScore of ContentFuzz is slightly lower than that of BERT-Attack and Reinforce-Attack, it remains within an acceptable range. This difference arises because BERTScore is computed from the similarity between embeddings of tokenized text. Unlike ContentFuzz, which rewrites the text, these adversarial attack methods preserve the positions of most tokens, a property that BERTScore favors because of positional encoding. However, because the substituted tokens are often nonsensical, the fluency of the generated adversarial examples degrades substantially, as indicated by their high perplexity. Overall, ContentFuzz demonstrates superior performance in generating effective and fluent content mutations, even when considered as a form of adversarial attack.

Analyzer	ASR	BERTScore	PPL	PPLr
Sem16				
BERT-Attack	0.371	0.934	1246.836	4.119
Reinforce-Attack	0.177	0.970	464.794	1.613
ContentFuzz	0.563	0.889	71.335	0.422
VAST				
BERT-Attack	0.679	0.959	81.2616	2.3789
Reinforce-Attack	0.191	0.995	38.772	1.055
ContentFuzz	0.883	0.878	10.041	0.322
C-STANCE-A				
Reinforce-Attack	0.003	0.916	886.696	2.758
ContentFuzz	0.910	0.752	21.242	0.164

Table 6: Comparison between ContentFuzz and adversarial attack methods on stance detection.

A.5 Case study

We present a case study in Table 7 to illustrate how ContentFuzz iteratively rewrites a post to change the prediction of the targeted stance analyzer. Our case is sampled from the Sem16 dataset (Mohammad et al., 2016) with the topic of Atheism. Starting from the original post in iteration 0, ContentFuzz gradually rewrites the post across iterations 1 and 2. The targeted stance analyzer initially predicts the original post as `Against` in iteration 0. After the first iteration of rewriting, the fuzzer generates a new post that is semantically equivalent to the original but with a lower confidence score. Using this new post as the seed, ContentFuzz further rewrites the post in iteration 2, which successfully flips the predicted stance to `Favor`, while maintaining the original meaning of the post.

Table 8 presents additional examples across all three datasets and both languages. The *Stance Change* column shows the ground-truth label and

[2] https://github.com/LinyangLee/BERT-Attack

Iteration	Stance	Confidence	Post
0	Against	1.0000	I am human. I look forward to the extinction of humanity with eager anticipation. We deserve nothing less.
1	Against	0.4582	I am human, and I eagerly await humanity's extinction. It's what we deserve.
2	Favor	0.4496	I am human, and I cannot wait for humanity's extinction. It's what we deserve.

Table 7: Case study illustrating confidence-guided content fuzzing across iterations. Topic: Atheism.

the analyzer's (incorrect) prediction (red). These examples illustrate several recurring patterns:

Surface register *vs*. underlying stance. In the Sem16 Feminist Movement and Legalization of Abortion examples, the rewrites soften the tone or formalize the register while preserving the core argument. For example, the Feminist Movement rewrite adopts a more measured register but advances the same pro-feminist argument, yet the model flips from `Favor` to `Against` with high confidence. Similarly, the Legalization of Abortion rewrite retains the original anti-abortion position in calmer phrasing, and BERT loses the `Against` signal.

Factually identical arguments. The VAST 3D Printing example shows that even when the rewrite preserves every factual claim (the same technical limitations in material variety and speed), BERT reverses its prediction from `Against` to `Favor`. This suggests that encoder-based models attend to phrasing cues (e.g., hedging constructions) rather than propositional content.

Zero-shot analyzer robustness. The VAST NATO example shows that LLM-based zero-shot analyzers are not immune: despite the rewrite retaining an explicit call to dissolve NATO, the analyzer shifts to `Neutral` after 13 iterations. The higher iteration count is consistent with the lower ESR of zero-shot analyzers reported in Section 5.1.

Cross-lingual transferability. The C-STANCE Conservative Groups example extends the case study to Chinese. The rewrite maintains the same favorable position on conservative groups, yet Chinese RoBERTa flips from `Favor` to `Neutral` with 0.98 confidence. This suggests that paraphrase-based content fuzzing generalizes across languages.

Human verification. To complement the computational metrics reported in the main paper, the authors and two independent PhD-level researchers from different domains independently reviewed all case study examples in Table 7 and Table 8. All reviewers agreed that the rewritten posts preserve the original core meaning, confirming that the semantic changes introduced by ContentFuzz are superficial rather than substantive.

Dataset	Stance Change	Original Post	Fuzzed Post
Sem16 *RoBERTa*	*Feminist Movement* Favor → Against Conf. 0.86	Expel them_ male millionaires from society! All Rights for women and children! Stop children women trafficking!	Let's champion the Feminist Movement by advocating for the rights of women and children, and by putting an end to the trafficking of vulnerable individuals.
Sem16 *BERT*	*Legalization of Abortion* Against → Neutral Conf. 0.67	@user1 @user2 @user3 Actually, child-murder is far worse these days. We live in more savage times.	In these difficult times, we are witnessing a disturbing increase in the loss of young lives.
VAST *BERT*	*3D Printing* Against → Favor Conf. 0.76	I can see 3D printing for prototypes, and some custom work. However manufacturing industries use thousands of plastics and thousands of metal alloys, few of which can be printed. I don't see that there is going to be a wholesale conversion to this relatively slow, materials-inflexible process.	While 3D printing is useful for prototypes and custom items, its broad use in manufacturing is still facing major challenges. The industry utilizes a huge range of plastics and metal alloys, but only a small percentage can currently be used with 3D printing. It's doubtful that this process will completely replace current methods due to its limitations in speed and material options.
VAST *Gemini*	*NATO* Against → Neutral Conf. 0.67	If we need to develop a world police force with other nations than we should have that conversation as to the how, the why and the cost. In the meantime it is long past time to draw NATO to a close.	Before we entertain the idea of a global police force, a deep dive into its practicalities, objectives, and costs is essential. Meanwhile, dissolving NATO is long overdue.
C-STANCE *RoBERTa*	*Conservative Groups* Favor → Neutral Conf. 0.98	浅析一下保守群体，其实不是一概而论。有些自由意志主义者，奉行不干预市场，但在社会价值观上允许堕胎和 LGBTQ。有些新保守主义者支持军事干预外部事务。而真保守派，持守新教信仰反堕胎反性别错乱，对外奉行不干预的孤立主义，支持低税不干预市场，强调社会秩序打击犯罪。	浅析保守群体，实际上并非铁板一块。比如，一些自由意志主义者信奉自由市场，却在堕胎和 LGBTQ 议题上持开放态度。另一些新保守主义者则倾向于在国际事务中采取军事干预。而传统保守派，则坚守新教信仰，反对堕胎和性别多元化，对外奉行不干预的孤立主义，支持低税收和自由市场，并着重强调社会秩序与打击犯罪。

Table 8: Case study examples across datasets, analyzers, and languages. The first column lists the dataset and target analyzer; the second column lists the topic, stance change (ground-truth → analyzer's incorrect prediction in red), and the analyzer's confidence on the fuzzed post.