

Social Media Guidance with GenAI Utilizing Jailbreaking Techniques

Md. Mahir Ashhab (ftm2nu), Radowan Mahmud Redoy (snf4za)

Abstract

Social media is a powerful platform for communication, especially during crises such as political unrest or natural disasters. However, posts are often misunderstood or removed due to violations of community guidelines, posing significant challenges for users in rural or under-served communities. In this work, we explore a lightweight approach using fine-tuned Large Language Models (LLMs) to automatically detect social media rule violations. We curate a synthetic dataset based on Facebook community standards and develop a structured prompt format to train the LLM. Our evaluation demonstrates that a small fine-tuned LLM can detect violations with high precision and provide human-readable explanations. Unlike previous work, our approach does not rely on neural network-based sentiment classification, focusing solely on regulatory comprehension through LLMs.

1 Introduction

Social media platforms such as Facebook, Twitter (now X), and TikTok play an increasingly central role in how information is disseminated during emergencies, social movements, and natural disasters. In countries with limited media infrastructure, such as Bangladesh, these platforms often serve as the primary means for grassroots communication, mutual aid coordination, and public awareness. However, with increased usage comes increased scrutiny, especially under the content moderation policies enforced by social media companies.

In 2024, Bangladesh witnessed two monumental events: the political instability that culminated in the collapse of the sitting government, and historic flooding that displaced hundreds of thousands of people. These crises led to surges in user-generated content—often emotionally charged, urgent, and written in informal language. We observed that numerous posts, despite their benevolent intent (e.g., calls for aid or expressing frustration), were

flagged or removed by automated moderation systems for allegedly violating platform rules. This disproportionately affected users from rural or low-literacy backgrounds who were unaware of community guidelines and lacked tools to verify the compliance of their messages before posting.

The root problem lies in the mismatch between how guidelines are interpreted by automated moderation models and how users—especially in the Global South—write and share content. While platforms use large-scale natural language models (NLP) for moderation, most end-users have no way to assess whether their posts might be misunderstood or flagged before publishing.

To address this gap, our project explores a lightweight, user-aligned approach using fine-tuned Large Language Models (LLMs) to proactively detect potential rule violations. We aim to offer an interpretable system that can act as a pre-check tool: helping users understand if a post may be flagged, why, and what specific guideline it might violate—without needing them to read hundreds of pages of policy documents.

Unlike previous research that primarily focuses on toxicity classification, sentiment analysis, or general-purpose red-teaming of LLMs, our work targets practical usability. We design a system that not only flags violations but also explains the logic behind the detection using natural language, closely emulating the format of platform guideline enforcement messages.

To achieve this, we:

- Curated a synthetic dataset of rule-violating and non-violating examples based on Facebook’s Community Standards.
- Designed prompt templates to encourage the LLM to produce consistent, interpretable outputs.
- Fine-tuned a small LLM to specialize in social media compliance checking.

- 082 • Evaluated the model on both structured met-
083 rics (e.g., F1 score) and qualitative inter-
084 pretability of its responses.

085 This report details the dataset construction,
086 prompt engineering, fine-tuning approach, model
087 behavior, comparative evaluation with alternative
088 models (e.g., LLaMA 3.2 1B), and our recom-
089 mendations for deploying such systems in real-world
090 contexts. We believe that even small LLMs, if
091 properly trained and prompted, can play a transfor-
092 mative role in making digital platforms safer and
093 more inclusive for marginalized voices.

094 2 Related Work

095 In recent years, the field of NLP has witnessed a
096 significant surge in research on aligning Large Lan-
097 guage Models (LLMs) with safety guidelines and
098 ethical norms. A prominent area of focus has been
099 jailbreak attacks—methods that circumvent safety
100 constraints built into LLMs—and the correspond-
101 ing defenses designed to mitigate such vulnerabili-
102 ties.

103 Xu et al. (2024) present a comprehensive taxon-
104 omy of jailbreak attacks and defense mechanisms
105 for large-scale models. Their work underscores
106 how prompt injection, adversarial paraphrasing,
107 and instruction misdirection can manipulate model
108 outputs, even in ostensibly aligned systems. While
109 these studies inform safety at the model develop-
110 ment level, they do not address the needs of end-
111 users trying to understand and avoid violating rules
112 in social media settings.

113 Similarly, Sharma et al. (2025) propose the con-
114 cept of *Constitutional Classifiers*, which aim to
115 protect LLMs from universal jailbreaks through
116 red-teaming at scale. Their approach aligns model
117 behavior with normative principles such as fairness,
118 non-violence, and non-discrimination. However,
119 the focus remains on upstream safeguards built into
120 the architecture or middleware layer, rather than on
121 post-hoc compliance analysis for user-generated
122 content.

123 In parallel, there has been substantial work
124 in content moderation using supervised classi-
125 fiers. Traditional methods employ rule-based
126 filtering, keyword blacklists, or statistical clas-
127 sifiers (e.g., logistic regression, SVMs) trained
128 on labeled datasets. While useful, these models
129 lack interpretability and often fail to capture nu-
130 ancanced violations. More recently, deep learning
131 methods—including convolutional neural networks

(CNNs), recurrent models, and transformer-based
132 classifiers—have been used for toxicity detection
133 (e.g., Jigsaw’s Perspective API), hate speech classi-
134 fication, and misinformation detection. However,
135 these systems rarely explain their decisions or adapt
136 to different platform-specific rule sets.

137 A related line of work includes studies on tox-
138 icity and sentiment analysis using social media
139 datasets such as Sentiment140, where emotion or
140 polarity of a post is mapped to positive, neutral, or
141 negative labels. While this helps gauge the tone of
142 content, it does not necessarily indicate whether a
143 post violates any policy.

144 Few-shot and instruction-tuned LLMs like GPT-
145 3.5, GPT-4, Claude, and LLaMA-2 have also been
146 shown to generalize well to classification tasks
147 when prompted effectively. However, in low-
148 resource or privacy-sensitive settings, these APIs
149 may not be viable due to cost, access, or data gov-
150 ernance concerns.

151 Our approach builds on these directions by tar-
152 geting a practical application: enabling social me-
153 dia users to pre-screen their own posts against com-
154 munity standards using a fine-tuned, open-weight
155 LLM. Unlike prior work, we combine prompt en-
156 gineering, synthetic dataset construction, and SFT-
157 based fine-tuning to create an interpretable viola-
158 tion detection system that generates not only clas-
159 sification labels but rule-aligned explanations.

160 This work is therefore positioned at the inter-
161 section of model alignment, content moderation,
162 and explainable AI for social impact—especially
163 in settings where access to platform guidance and
164 digital literacy is limited.

166 3 Methodology

167 Our system for detecting community guideline vio-
168 lations is built on top of a small open-weight causal
169 LLM, fine-tuned using a custom dataset of syn-
170 thetically generated examples. The methodology
171 consists of three core components: data prepara-
172 tion, prompt design, and model fine-tuning. The
173 following subsections describe each stage in detail.

174 3.1 Data Collection and Preparation

175 We manually curated a dataset based on Facebook’s
176 publicly available *Community Standards*, which
177 outline the kinds of content prohibited on the plat-
178 form. These include but are not limited to: hate
179 speech, scam promotions, blackmail, threats of vio-
180 lence, spam, impersonation, and misleading finan-

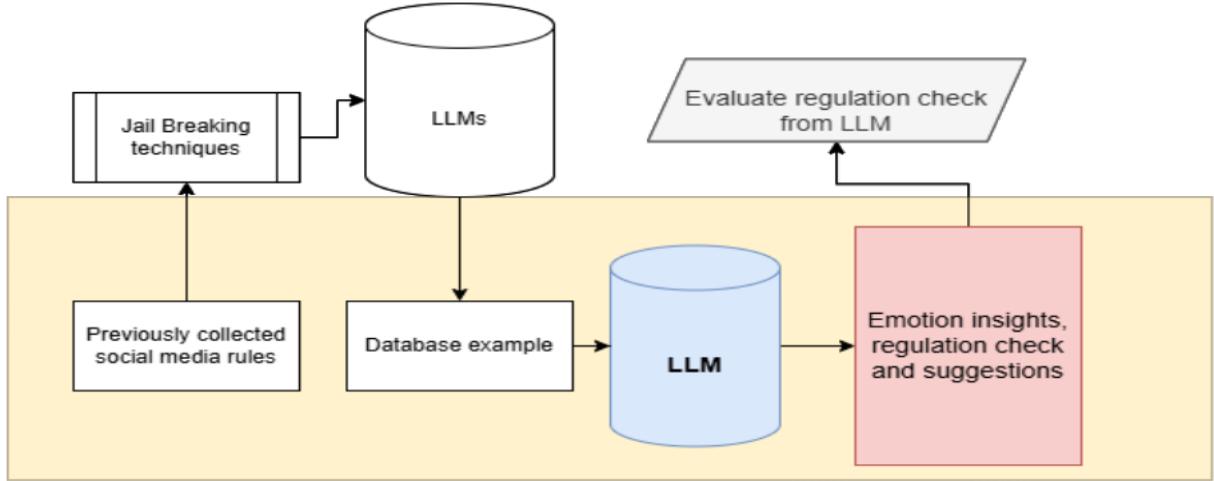


Figure 1: High-level overview of the proposed workflow

cial claims.

Since no publicly available dataset directly maps social media posts to these specific rule categories with explanations, we created synthetic examples that mimic the tone and structure of real-world violations. For each example, we ensured linguistic diversity by including informal language, short and long-form text, imperative statements, and emotionally charged content.

Each sample is structured as a dictionary object containing the following fields:

- **Input Text:** A short social media-style sentence or post.
- **Violation Flag:** A boolean value indicating whether the content violates a community guideline.
- **Rule Name:** The specific Facebook guideline (e.g., “Advance Fee Loan Scams”) being violated.
- **Explanation:** A natural language rationale explaining the nature of the violation.
- **ID:** A unique identifier used for logging and evaluation purposes.

Here is an example of a data point:

```
{
  "input": "Get instant approval on your $5000 loan! Just pay a $200 processing fee upfront.",
  "output": {
    "violation": true,
    "rule": "Offers loans requiring the user to pay an advance fee",
    "explanation": "This constitutes an advance fee loan scheme, which violates platform policy."
  }
}
```

213 "explanation": "This constitutes an
214 advance fee loan scheme, which
215 violates platform policy."
216 },
217 "id": 0
218 }

3.2 Prompt Design

To ensure that the model produces consistent, structured responses, we designed a specialized instruction-style prompt template. The goal was to elicit outputs that not only indicate whether a rule is violated, but also provide justification in natural language.

The standard prompt format is:

227 If the given sentence
228 “{sentence}” is posted in any
229 social media, will it violate
230 any rule? If violation is true
231 then which rule is violated and
232 what is the explanation. Your
233 answer must strictly include if
234 violation is true or false.

This prompt forces the model to engage in structured reasoning and produce three required components: a boolean decision, a rule label (if applicable), and a short explanation. During training, the model was supervised on thousands of such prompt-output pairs, helping it internalize rule logic and response formatting.

This strategy mimics prompt engineering methods seen in instruction-tuned LLMs like GPT-4 or Claude, but adapted for a fine-tuned deployment setting with a smaller model and fixed prompt style.

	<h3>3.3 Model Fine-Tuning</h3> <p>We fine-tuned a small causal decoder-only LLM (7B or smaller) using Supervised Fine-Tuning (SFT) with the Hugging Face transformers and tr1 libraries. The base model was selected for its compact size, open licensing, and compatibility with parameter-efficient training techniques.</p> <p>Fine-tuning was performed on a workstation with limited GPU resources, so we adopted a configuration optimized for memory usage:</p> <ul style="list-style-type: none"> • Learning Rate: 2e-4 — a moderately aggressive rate suitable for short training cycles. • Weight Decay: 0.001 — to prevent overfitting on small datasets. • Warmup Ratio: 0.03 — gradual learning rate increase during initial steps. • Max Gradient Norm: 0.3 — to clip excessively large gradients. • Gradient Accumulation Steps: 1 — compatible with our batch size and hardware limits. • Epochs: 1 — as the dataset is synthetic and cleanly structured. <p>Tokenization was handled using the model’s default tokenizer, with sequence length capped at 512 tokens per sample. Mixed precision training (FP16) was enabled to reduce GPU memory usage. No external datasets were used for pretraining or augmentation to ensure full control over input-output mappings and to prevent model behavior drift.</p> <p>The resulting fine-tuned model outputs JSON-style responses with high consistency and was tested against a manually curated validation set as described in the next section.</p>		4.1 Qualitative Examples The fine-tuned model was able to generate outputs that aligned well with the structure and content of our training prompts. Below are two examples taken from our evaluation set: Example 1: <ul style="list-style-type: none"> • Input: “You must use this bribe or otherwise you are done.” • Model Output: <ul style="list-style-type: none"> – Violation: True – Rule: Bribery or blackmail threats – Explanation: The sentence explicitly refers to the use of a bribe in a threatening context, which qualifies as blackmail and violates platform rules against coercive behavior. Example 2: <ul style="list-style-type: none"> • Input: “I will see how far you can go.” • Model Output: <ul style="list-style-type: none"> – Violation: True – Rule: Threatening or advocating violence – Explanation: The phrasing suggests a veiled threat of escalation, violating platform guidelines on non-violence and intimidation. <p>These examples illustrate the model’s ability to:</p> <ul style="list-style-type: none"> • Extract the underlying intent behind ambiguous or implicit threats. • Map behaviors to specific guideline categories. • Provide natural language explanations that mirror the reasoning a human moderator might offer.
	<h3>4 Results and Evaluation</h3> <p>This section presents both qualitative and quantitative results of our fine-tuned LLM. We evaluate the system based on two main criteria:</p> <ul style="list-style-type: none"> • Correctness and clarity of its classification and explanation outputs. • Metric-based performance on a hand-validated test set. 		4.2 Quantitative Evaluation We evaluated the model on a validation set of 32 manually crafted and balanced samples that were not seen during training. The following confusion matrix was observed:
	<ul style="list-style-type: none"> • True Positives (TP): 20 • True Negatives (TN): 8 		326 327

- False Positives (FP): 2
- False Negatives (FN): 2

Using these, we computed the following metrics:

- **Precision:** 90.91%
- **Recall:** 90.91%
- **F1 Score:** 90.91%
- **Accuracy:** 87.5%

These results indicate that the model not only identifies violations reliably but also minimizes misclassifications. The balanced F1 score confirms that the model does not exhibit strong bias toward either class, a common problem in moderation systems that favor over-flagging or under-flagging content.

4.3 Comparison with LLaMA 3.2 1B

To assess whether smaller models could match the performance of our primary fine-tuned LLM, we conducted the same experiment using the **LLaMA 3.2 1B** model with identical data and prompts.

Unfortunately, the model performed poorly:

- In multiple cases, it produced **no output**—returning empty strings or failing to complete responses entirely.
- When it did respond, it often failed to follow the JSON-style prompt format, omitting required fields such as the violation flag or the explanation.
- It frequently hallucinated rules not found in the training set and produced vague or redundant explanations such as “This is bad” or “Not allowed,” without reference to specific violations.

We attribute these failures to:

1. **Insufficient model capacity:** At 1B parameters, LLaMA 3.2 lacks the depth required to internalize both the logical structure and language complexity of the task.
2. **Poor instruction following:** The base model was not trained with alignment objectives (e.g., RLHF or instruction tuning), making it unable to conform to prompt constraints.

3. **Generation instability:** Certain prompts failed to trigger completions altogether, even with adjusted decoding settings like temperature and top-p.

Conclusion: While LLaMA 3.2 1B is useful for lightweight inference, it was not suitable for structured moderation tasks involving detailed reasoning and compliance interpretation. As such, it was excluded from our final evaluations.

5 Discussion

The evaluation results confirm that a lightweight, fine-tuned LLM can achieve strong performance in identifying guideline violations and providing coherent explanations. Our design approach—based on prompt alignment, rule-specific labeling, and structured output—makes the system practical for real-world deployment in settings with limited resources.

5.1 Strengths of the System

- **Interpretability:** One of the core advantages of our approach is that the model does not simply produce a binary label but explains why a post violates a rule. This improves user trust and enables actionable feedback.
- **Flexibility:** The system is not hard-coded to a specific platform’s enforcement API. By re-training on a different guideline set, it could be adapted to platforms like Reddit, Twitter, or TikTok.
- **Lightweight Deployment:** Because we used a compact LLM with only one epoch of fine-tuning, our model can be deployed locally or on low-cost servers, making it accessible for NGOs or small teams working in digital safety and civic tech.

- **Robust Prompting:** The carefully constructed instruction-style prompts encouraged the model to produce reliable and explainable outputs consistently—even for linguistically diverse inputs.

5.2 Limitations

Despite its strengths, our system has several known limitations that affect its generalization to open-domain, user-generated content:

- 413 1. **Synthetic Dataset Bias:** Because our dataset
 414 was synthetically constructed, it does not cap-
 415 ture the full diversity and ambiguity of real-
 416 world social media language. Code-switching,
 417 sarcasm, emojis, and slang are underrepre-
 418 sented.
- 419 2. **Binary Framing:** The model is currently con-
 420 strained to produce a binary classification (vi-
 421 olation or not), but in real moderation scenar-
 422 ios, content may fall into gray areas or require
 423 human review.
- 424 3. **No Multilingual Support:** Our model is
 425 English-only. This excludes a significant por-
 426 tion of users in Bangladesh and elsewhere
 427 who post in Bengali or mixed languages.
- 428 4. **Hardcoded Rule Mapping:** The model does
 429 not learn rules from scratch but rather maps
 430 inputs to predefined rule categories. This lim-
 431 its scalability to platforms with dynamic or
 432 overlapping guidelines.
- 433 **5.3 Ethical and Societal Considerations**
- 434 Using LLMs for content moderation carries both
 435 promise and risk. While models can reduce false
 436 positives and provide transparency, they may also
 437 encode unintended biases or produce overconfident
 438 outputs that users take at face value.
- 439 Our system is designed to be advisory—not puni-
 440 tive. It does not automatically flag or delete content
 441 but informs the user about potential risks. This
 442 ensures that decision-making remains in human
 443 hands, respecting freedom of expression.
- 444 Moreover, interpretability plays a critical role
 445 in preserving user dignity. Marginalized users of-
 446 ten feel alienated when content is removed without
 447 explanation. A model that “speaks back” with rea-
 448 sons helps users learn and adapt, building digital
 449 resilience.
- 450 **5.4 Lessons Learned**
- 451 Through this project, we observed that:
- 452 • Small LLMs can outperform expectations
 453 when fine-tuned with aligned objectives and
 454 structured data.
 - 455 • Prompt design is just as important as dataset
 456 quality—many early failures were due to
 457 vague instructions, not poor modeling.
- Explainability is a feature, not an afterthought,
 459 and should be built into the architecture of
 460 moderation systems from the beginning.
- 461 **6 Conclusion and Future Work**
- 462 **6.1 Conclusion**
- 463 In this work, we proposed a lightweight, inter-
 464 pretable, and customizable framework for detect-
 465 ing social media rule violations using fine-tuned
 466 Large Language Models. Motivated by challenges
 467 observed during social crises in Bangladesh, our ap-
 468 proach empowers users—especially those in rural
 469 or underrepresented communities—to pre-screen
 470 their content for guideline violations before posting.
- 471 We curated a rule-aligned synthetic dataset mod-
 472 eled after Facebook’s Community Standards and
 473 developed structured prompts to elicit explainable
 474 outputs from a causal LLM. After supervised fine-
 475 tuning, the model demonstrated high accuracy
 476 (87.5%) and F1 score (90.91%) on a hand-labeled
 477 validation set, while providing rule-specific rea-
 478 soning for each decision. Compared to a smaller
 479 LLaMA 3.2 1B model, which failed to generate inter-
 480 pretable or consistent outputs, our chosen model
 481 proved far more reliable for compliance tasks.
- 482 By focusing on transparency, we designed the
 483 model to communicate not just *what* content is
 484 problematic, but *why*, closing the feedback loop for
 485 users unfamiliar with complex moderation policies.
 486 This approach bridges the gap between centralized
 487 AI moderation infrastructure and grassroots digital
 488 empowerment.
- 489 **6.2 Future Work**
- 490 While our results are promising, several key direc-
 491 tions remain for future exploration:
- 492 1. **Real-World Data Integration:** To improve
 493 generalization, future versions should be
 494 trained and evaluated on real flagged posts
 495 from platforms like Facebook or Reddit (pend-
 496 ing ethical clearance and privacy safeguards).
 - 497 2. **Multilingual and Code-Switched Texts:**
 498 Given that many users in the Global South
 499 write in mixed English and native languages,
 500 future models should support multilingual
 501 inputs, starting with Bengali-English code-
 502 switched text.
 - 503 3. **Multi-Class and Uncertainty Outputs:** Cur-
 504 rent outputs are binary; future versions could

506 use confidence thresholds or produce multi-
507 label classifications, including “borderline”
508 cases or “manual review required” tags.

509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551

5. NousResearch. (2024). *LLaMA-2-7B-Chat-HF model*. Hugging Face

509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551

4. **User Feedback Loop:** We aim to design an interactive web demo where users can test posts, receive suggestions, and optionally correct flagged content. This would enable continual human-in-the-loop learning.

514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551

5. **Quantization and Mobile Deployment:** To make the system deployable in low-connectivity environments, we plan to explore quantization-aware training, QLoRA, or distillation methods for on-device inference.

514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551

6. **Expanded Rule Coverage:** Our current taxonomy includes 8–10 rule types. Incorporating broader categories (e.g., misinformation, spam links, impersonation) would make the system more comprehensive and platform-agnostic.

525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551

Ultimately, we believe that fine-tuned LLMs—if trained and aligned properly—can function as educational and protective tools, not just moderation engines. By offering users meaningful explanations and the opportunity to learn from their mistakes, we hope to support more inclusive and fair participation in the digital public sphere.

532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551

Repository and Code: All experiments, training code, and synthetic datasets are available at: <https://github.com/ashhab7/genAI/tree/master>

7 References

1. Z. Xu, Y. Liu, G. Deng, Y. Li, S. Picek. (2024). *A Comprehensive Study of Jailbreak Attack versus Defense for Large Language Models*. arXiv:2402.13457
2. M. Sharma et al. (2025). *Constitutional Classifiers: Defending against Universal Jailbreaks across Thousands of Hours of Red Teaming*. arXiv:2501.18837
3. A. Go, R. Bhayani, L. Huang. (2009). *Twitter Sentiment Classification using Distant Supervision*.
4. E. Hu et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv:2106.09685