

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.

- Evaluated the model on both structured metrics (e.g., F1 score) and qualitative interpretability of its responses.

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

2 Related Work

In recent years, the field of NLP has witnessed a significant surge in research on aligning Large Language Models (LLMs) with safety guidelines and ethical norms. A prominent area of focus has been jailbreak attacks—methods that circumvent safety constraints built into LLMs—and the corresponding defenses designed to mitigate such vulnerabilities.

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

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

In parallel, there has been substantial work in content moderation using supervised classifiers. Traditional methods employ rule-based filtering, keyword blacklists, or statistical classifiers (e.g., logistic regression, SVMs) trained on labeled datasets. While useful, these models lack interpretability and often fail to capture nuanced violations. More recently, deep learning methods—including convolutional neural networks

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

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

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

Our approach builds on these directions by targeting a practical application: enabling social media users to pre-screen their own posts against community standards using a fine-tuned, open-weight LLM. Unlike prior work, we combine prompt engineering, synthetic dataset construction, and SFT-based fine-tuning to create an interpretable violation detection system that generates not only classification labels but rule-aligned explanations.

This work is therefore positioned at the intersection of model alignment, content moderation, and explainable AI for social impact—especially in settings where access to platform guidance and digital literacy is limited.

3 Methodology

Our system for detecting community guideline violations is built on top of a small open-weight causal LLM, fine-tuned using a custom dataset of synthetically generated examples. The methodology consists of three core components: data preparation, prompt design, and model fine-tuning. The following subsections describe each stage in detail.

3.1 Data Collection and Preparation

We manually curated a dataset based on Facebook’s publicly available *Community Standards*, which outline the kinds of content prohibited on the platform. These include but are not limited to: hate speech, scam promotions, blackmail, threats of violence, spam, impersonation, and misleading finan-

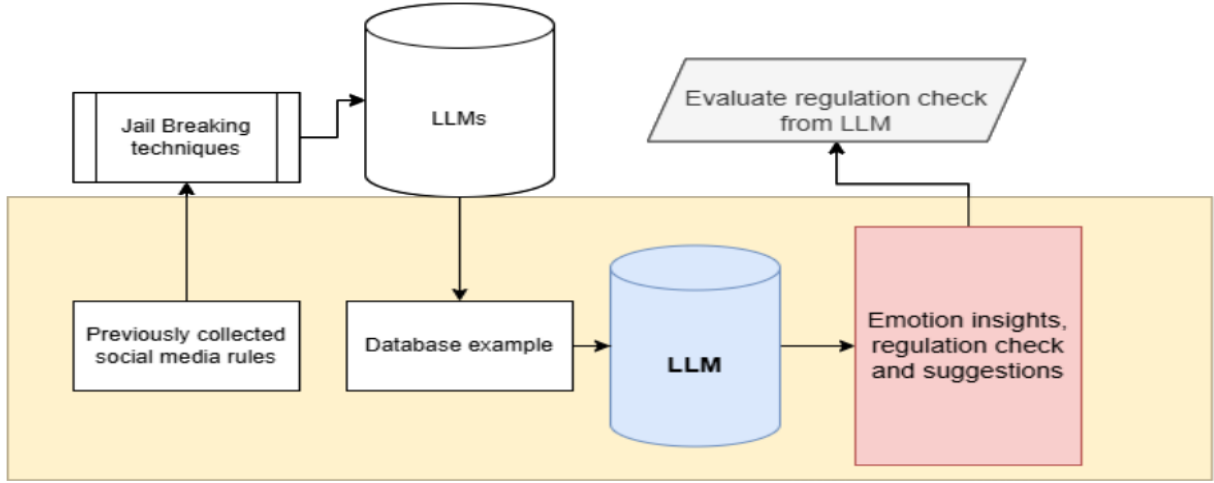


Figure 1: High-level overview of the proposed workflow

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."
  },
  "id": 0
}

```

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:

```

If the given sentence
"{sentence}" is posted in any
social media, will it violate
any rule? If violation is true
then which rule is violated and
what is the explanation. Your
answer must strictly include if
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.

3.3 Model Fine-Tuning

We fine-tuned a small causal decoder-only LLM (7B or smaller) using Supervised Fine-Tuning (SFT) with the Hugging Face transformers and trl libraries. The base model was selected for its compact size, open licensing, and compatibility with parameter-efficient training techniques.

Fine-tuning was performed on a workstation with limited GPU resources, so we adopted a configuration optimized for memory usage:

- **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.

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.

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.

4 Results and Evaluation

This section presents both qualitative and quantitative results of our fine-tuned LLM. We evaluate the system based on two main criteria:

- **Correctness and clarity** of its classification and explanation outputs.
- **Metric-based performance** on a hand-validated test set.

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:

- **Input:** “You must use this bribe or otherwise you are done.”
- **Model Output:**
 - **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:

- **Input:** “I will see how far you can go.”
- **Model Output:**
 - **Violation:** True
 - **Rule:** Threatening or advocating violence
 - **Explanation:** The phrasing suggests a veiled threat of escalation, violating platform guidelines on non-violence and intimidation.

These examples illustrate the model’s ability to:

- 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.

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:

- True Positives (TP): 20
- True Negatives (TN): 8

328	• False Positives (FP): 2	369
329	• False Negatives (FN): 2	370
330	Using these, we computed the following metrics:	371
331	• Precision: 90.91%	372
332	• Recall: 90.91%	373
333	• F1 Score: 90.91%	374
334	• Accuracy: 87.5%	375
335	These results indicate that the model not only	376
336	identifies violations reliably but also minimizes	377
337	misclassifications. The balanced F1 score confirms	378
338	that the model does not exhibit strong bias toward	379
339	either class, a common problem in moderation sys-	380
340	tems that favor over-flagging or under-flagging con-	381
341	tent.	382
342	4.3 Comparison with LLaMA 3.2 1B	383
343	To assess whether smaller models could match the	384
344	performance of our primary fine-tuned LLM, we	385
345	conducted the same experiment using the LLaMA	386
346	3.2 1B model with identical data and prompts.	387
347	Unfortunately, the model performed poorly:	388
348	• In multiple cases, it produced no out-	389
349	put —returning empty strings or failing to	390
350	complete responses entirely.	391
351	• When it did respond, it often failed to follow	392
352	the JSON-style prompt format, omitting re-	393
353	quired fields such as the violation flag or	394
354	the explanation.	395
355	• It frequently hallucinated rules not found in	396
356	the training set and produced vague or redun-	397
357	dant explanations such as “This is bad” or	398
358	“Not allowed,” without reference to specific	399
359	violations.	400
360	We attribute these failures to:	401
361	1. Insufficient model capacity: At 1B param-	402
362	eters, LLaMA 3.2 lacks the depth required	403
363	to internalize both the logical structure and	404
364	language complexity of the task.	405
365	2. Poor instruction following: The base model	406
366	was not trained with alignment objectives	407
367	(e.g., RLHF or instruction tuning), making	408
368	it unable to conform to prompt constraints.	409
	3. Generation instability: Certain prompts	410
	failed to trigger completions altogether, even	411
	with adjusted decoding settings like tempera-	412
	ture and top-p.	
	Conclusion: While LLaMA 3.2 1B is useful	
	for lightweight inference, it was not suitable for	
	structured moderation tasks involving detailed rea-	
	soning 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 co-	
	herent 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 re-	
	sources.	
	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 con-	
	structed 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:	

1. **Synthetic Dataset Bias:** Because our dataset was synthetically constructed, it does not capture the full diversity and ambiguity of real-world social media language. Code-switching, sarcasm, emojis, and slang are underrepresented.
2. **Binary Framing:** The model is currently constrained to produce a binary classification (violation or not), but in real moderation scenarios, content may fall into gray areas or require human review.
3. **No Multilingual Support:** Our model is English-only. This excludes a significant portion of users in Bangladesh and elsewhere who post in Bengali or mixed languages.
4. **Hardcoded Rule Mapping:** The model does not learn rules from scratch but rather maps inputs to predefined rule categories. This limits scalability to platforms with dynamic or overlapping guidelines.

5.3 Ethical and Societal Considerations

Using LLMs for content moderation carries both promise and risk. While models can reduce false positives and provide transparency, they may also encode unintended biases or produce overconfident outputs that users take at face value.

Our system is designed to be advisory—not punitive. It does not automatically flag or delete content but informs the user about potential risks. This ensures that decision-making remains in human hands, respecting freedom of expression.

Moreover, interpretability plays a critical role in preserving user dignity. Marginalized users often feel alienated when content is removed without explanation. A model that “speaks back” with reasons helps users learn and adapt, building digital resilience.

5.4 Lessons Learned

Through this project, we observed that:

- Small LLMs can outperform expectations when fine-tuned with aligned objectives and structured data.
- Prompt design is just as important as dataset quality—many early failures were due to vague instructions, not poor modeling.

- Explainability is a feature, not an afterthought, and should be built into the architecture of moderation systems from the beginning.

6 Conclusion and Future Work

6.1 Conclusion

In this work, we proposed a lightweight, interpretable, and customizable framework for detecting social media rule violations using fine-tuned Large Language Models. Motivated by challenges observed during social crises in Bangladesh, our approach empowers users—especially those in rural or underrepresented communities—to pre-screen their content for guideline violations before posting.

We curated a rule-aligned synthetic dataset modeled after Facebook’s Community Standards and developed structured prompts to elicit explainable outputs from a causal LLM. After supervised fine-tuning, the model demonstrated high accuracy (87.5%) and F1 score (90.91%) on a hand-labeled validation set, while providing rule-specific reasoning for each decision. Compared to a smaller LLaMA 3.2 1B model, which failed to generate interpretable or consistent outputs, our chosen model proved far more reliable for compliance tasks.

By focusing on transparency, we designed the model to communicate not just *what* content is problematic, but *why*, closing the feedback loop for users unfamiliar with complex moderation policies. This approach bridges the gap between centralized AI moderation infrastructure and grassroots digital empowerment.

6.2 Future Work

While our results are promising, several key directions remain for future exploration:

1. **Real-World Data Integration:** To improve generalization, future versions should be trained and evaluated on real flagged posts from platforms like Facebook or Reddit (pending ethical clearance and privacy safeguards).
2. **Multilingual and Code-Switched Texts:** Given that many users in the Global South write in mixed English and native languages, future models should support multilingual inputs, starting with Bengali-English code-switched text.
3. **Multi-Class and Uncertainty Outputs:** Current outputs are binary; future versions could

506
507
508

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

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

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.

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](https://arxiv.org/abs/2402.13457)

2. M. Sharma et al. (2025). *Constitutional Classifiers: Defending against Universal Jailbreaks across Thousands of Hours of Red Teaming*. [arXiv:2501.18837](https://arxiv.org/abs/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](https://arxiv.org/abs/2106.09685)

5. NousResearch. (2024). *LLaMA-2-7B-Chat-HF model*. [Hugging Face](https://huggingface.co/NousResearch/LLaMA-2-7B-Chat-HF)

550
551