Project 3: Toxicity Patterns in Open Discussions

Alif Al Hasan
Case Western Reserve University
axh1218@case.edu

This project investigates toxicity patterns across two major programming support platforms: Reddit's r/learnprogramming and GitHub bug-report discussions—using 16,266 Reddit comments and 9,000 GitHub issue comments. I compared two automated detection approaches: *Detoxify* and *Llama-3.3-70B*.

1 INTRODUCTION

1.1 Background and Motivation

Online programming communities are central to how developers learn and collaborate, yet platforms differ greatly in purpose and communication norms. r/learnprogramming serves as an informal learning space for novices, while GitHub issues function as professional, hierarchical project-management channels. These contextual differences shape expectations: casual encouragement common on Reddit would be inappropriate in GitHub's professional environment.

Toxic interactions in such settings discourage participation, particularly among newcomers and underrepresented groups, and can undermine both learning and open-source development. Because help-seeking involves vulnerability, dismissive or hostile responses can have long-term negative effects. As programming education and open-source activity increasingly move online, sustaining healthy community interactions becomes essential.

Despite these concerns, toxicity in programming communities remains understudied. Existing detection tools are typically trained on general social media and may misinterpret technical language, overlook subtle dismissiveness, or over-flag casual encouragements. This limits effective moderation and highlights the need for platform-aware analysis.

1.2 Project Scope and Contributions

This project provides a systematic, practical examination of toxicity across Reddit and GitHub, focusing on detection performance, platform differences, and methodological insights. The main contributions are:

- **Cross-platform comparative analysis:** Evaluated Detoxify and Llama-3.3-70B on both Reddit and GitHub using consistent methodology.
- **Limitations of general-purpose detectors:** Detoxify shows only 22.5% accuracy on GitHub, demonstrating significant constraints for use in technical communities.
- **Context-dependent LLM performance:** Llama reaches 72.5% accuracy on GitHub, showing that even advanced LLM-based methods require platform-specific tuning.
- **Platform-specific toxicity taxonomies:** Through qualitative analysis, identified distinct toxicity patterns—social dismissiveness and gatekeeping on Reddit versus technical dismissiveness and professional hostility on GitHub.
- **Methodological insights:** Demonstrated how cultural factors create divergent detection challenges.

2 DATASETS

To study toxicity across platforms, I used two datasets that represent different types of programming support: a social learning space (Reddit) and a professional technical collaboration space (GitHub).

2.1 Dataset 1: Reddit r/learnprogramming

2.1.1 Data Source and Collection. Reddit's r/learnprogramming mainly focused on helping beginners. The subreddit encourages friendly, patient behaviour and avoids discouraging or rude replies. This dataset comes from an ongoing research project that studies emotional expressions in novice help-seeking posts.

They collected all posts from January 2023 to November 2024 (around 83,000 posts). They then identified posts containing learning-centred emotions (LCEs): anxiety, boredom, confusion, curiosity, delight, engagement, frustration, and surprise. After LLM filtering and manual checking, 1,500 posts with confirmed emotional content and from beginners were selected.

For this work, I collected all comments on these 1,500 emotional posts. This gave data that captures how the community responds to beginners who are already showing some emotional vulnerability, which is important for understanding toxicity in learning situations.

2.1.2 Dataset Characteristics. The Reddit dataset includes 16,266 which show how community members respond to beginners who are confused, frustrated, anxious, or curious. Each post also has its LCE label, allowing us to see how emotional context may influence the tone of replies.

2.1.3 Preprocessing.

- Removed all AutoModerator comments, since they are automated and not part of real interactions.
- Removed comments shorter than 10 characters, as these usually contain too little information for toxicity analysis.
- Linked each comment to its parent post.

2.2 Dataset 2: GitHub Bug Report Discussions

2.2.1 Data Source and Collection. The GitHub dataset comes from the work of Imran et al. [1], who studied silent toxicity in bug report discussions. They collected issue comments from 100 popular open-source repositories based on strict selection criteria:

(1) At least 1,000 stars (to ensure active, well-used projects).
(2) Active development activity in 2024 (up to September 5).
(3) Clear signs of moderation such as Codes of Conduct or locked issues.
(4) Bug-related labels (bug, defect, crash, triage, fix).
(5) English-language issue comments.

2.2.2 Dataset Characteristics. The full GitHub dataset contains 91,929 comments across 8,723 issues. For our analysis, we used a 10% subset of 9,000 comments to reduce computational cost while keeping diversity across repositories and contributor types.

A key feature of GitHub data is the author_association field, which indicates the user's role in the repository: OWNER, MEMBER, CONTRIBUTOR, FIRST_TIME_CONTRIBUTOR, or NONE. This helps us study how roles and power differences affect communication. The GitHub context is more professional compared to Reddit. Users communicate using their real identities or professional profiles, and discussions focus on specific technical issues.

The dataset includes 206 manually labeled comments identified by the original authors. These comments show behaviours such as dismissiveness, condescension, and unprofessional tone, and serve as ground truth for evaluating detection models.

2.2.3 Preprocessing. I applied a preprocessing pipeline similar to Reddit:

- Removed bot-generated messages (e.g., dependabot, github-actions, codecov-bot).
- Removed comments shorter than 10 characters.

2.3 Cross-Platform Dataset Comparison

Table 1 highlights the main differences between Reddit and GitHub that influence communication style and toxicity patterns.

Table 1: Platform Dataset Comparison

Characteristic	Reddit	GitHub
Context	Social learning	Technical problem-solving
Primary users	Beginners	Developers
Emotional content	High	Low
Goal of interaction	Learning support	Bug resolution
Anonymity	High	Low (professional identity)
Sample size	16,266 comments	9,000 comments

These differences strongly influence how toxicity appears on each platform. Reddit encourages emotional expression and supportive communication, while GitHub expects professional, goal-oriented discussion. Power dynamics are also very different, with GitHub maintainers having much more authority over discussions. Because of these contrasting environments, the same sentence may be acceptable on one platform but considered toxic on the other. This makes cross-platform analysis important for understanding how toxicity should be detected and interpreted.

3 METHODOLOGY

My methodology consists of two main components: (1) applying two complementary toxicity detection approaches, and (2) evaluating and comparing method performance across Reddit and GitHub.

3.0.1 Reddit Annotation Procedure. Reddit contains 16,266 comments, making full manual annotation infeasible. We therefore adopted a two-stage filtering-and-review approach. I applied both toxicity detection methods (Detoxify and Llama) and received toxicity score from 0-10 for all comments.

Comments scoring at least 7 under either method were flagged for closer inspection. This produced 170 comments. This strategy balances practicality and coverage: although it cannot guarantee detection of all toxic content, it prioritizes comments most likely to contain toxic or discouraging language.

Each flagged comment was manually reviewed in context, including its parent post and emotion label. During annotation, we distinguished between:

- supportive profanity ("get in the fucking water and learn"),
- genuinely dismissive, discouraging, or hostile remarks ("why are you even programming").

Of the 170 flagged comments, approximately 25 were judged to be genuinely toxic. This yields a true-positive rate of roughly 15%, with the remaining comments primarily reflecting false positives caused by profanity, sarcasm, or misunderstood context. Because unflagged comments were not exhaustively reviewed, recall for Reddit cannot be precisely calculated, but the manually labeled subset enables qualitative error analysis and approximate performance estimation.

3.1 Toxicity Detection Approaches

I evaluated two complementary toxicity detection methods: a conventional classifier (Detoxify) and a context-aware large language model (Llama). This dual-method setup allows us to study both lexical and context-sensitive detection behaviors.

3.1.1 Detoxify. Detoxify is a transformer-based toxicity classifier trained on Wikipedia and social media datasets. We use the `original` model, which outputs probability scores for several toxicity dimensions. Detoxify was applied unchanged to all Reddit and GitHub comments, serving as a baseline to assess how well general-purpose toxicity detectors transfer to technical communities.

Advantages:

- free, fast, and computationally lightweight
- deterministic and reproducible
- widely used baseline model

Limitations:

- trained on non-technical content, often missing nuances of technical dismissiveness
- relies heavily on lexical cues such as profanity
- cannot incorporate platform-specific or domain-specific context

3.1.2 Llama-3.3-70B-Instruct. To capture context-sensitive toxicity, I used Llama-3.3-70B-Instruct with a custom prompt. The prompt defines toxicity using a detailed rubric:

- 0–3: neutral or supportive
- 4–6: condescending, unhelpful, or harsh
- 7–8: mocking, elitist, or clearly rude
- 9–10: extremely hostile or explicitly offensive

Advantages:

- understands context and intent, not just keywords
- adaptable via prompting
- capable of recognizing subtle, non-explicit toxicity

Limitations:

- significantly higher computational cost
- potential variability in outputs

- requires careful prompt design

3.2 Evaluation Strategy

My evaluation focuses on three dimensions: within-platform performance, cross-platform comparison, and qualitative pattern analysis.

3.2.1 Within-Platform Performance. Using the 80 ground-truth toxic comments, we compute recall:

$$\text{Recall} = \frac{\text{Number of toxic comments detected}}{80}.$$

Precision cannot be determined due to lack of labels for the remaining comments. For reddit data, I did not find any ground truths.

3.2.2 Cross-Platform Comparison. I compared methods across platforms to identify:

- performance differences in technical vs. social media contexts,
- types of toxicity each method misses or overflags,
- agreement and divergence patterns between methods.

3.2.3 Qualitative Pattern Analysis. Finally, I conducted qualitative analysis of toxic and borderline comments from both platforms. This includes categorizing examples, identifying linguistic markers of toxicity, and explaining method-specific error patterns. These insights help interpret quantitative findings and illuminate how toxicity manifests differently in programming communities.

4 RESULTS

I presented findings across both platforms, cross-platform comparison, and qualitative toxicity patterns.

4.1 GitHub Performance: Llama Strongly Outperforms Detoxify

Table 2 shows performance on 80 manually labeled GitHub toxic comments.

Table 2: GitHub Detection Performance

Method	Detected	Recall
Detoxify	18/80	22.5%
Llama-3.3-70B	58/80	72.5%
Both methods	18	22.5%
Only Llama	40	50.0%
Neither	22	27.5%

Llama achieves 72.5% accuracy (3.2× higher than Detoxify). Every Detoxify detection is a subset of Llama's, indicating Detoxify does not capture any unique toxicity types. Missed cases typically involve subtle professional dismissiveness (e.g., "not a bug," "duplicate, closing") that lack explicit hostile markers.

4.2 Reddit Analysis

Manual review of Detoxify-flagged Reddit comments shows almost similar performance (Table 3).

In Reddit dataset, both the models performed almost similarly in identifying toxic comments.

Table 3: Reddit Detection Analysis (170 Flagged)

Category	Count	Percentage
Flagged	170	100%
Actually toxic	~25	14.7%
False positives	~145	85.3%

4.2.1 False Positive Examples. Detoxify consistently assigns extreme toxicity scores (9–10) to supportive comments containing profanity for emphasis, humor, or enthusiasm (Table 4). Llama correctly identifies them as non-toxic.

Table 4: Reddit False Positives: Helpful Comments Misclassified

Comment	Detoxify	Llama
"get in the fucking water and learn"	10	0
"holy shit now i get it"	9	0
"if you fuck up memory management..."	10	0

4.2.2 Actual Toxic Reddit Comments. Among the 25 genuinely toxic comments, some samples: (Table 5).

4.3 Cross-Platform Comparison

Performance changes across platforms: Detoxify fails on GitHub but performs quite good on Reddit; Llama showed balanced performance across both datasets. This demonstrates that toxicity detection is strongly dependent on platform-specific linguistic norms.

4.4 Qualitative Toxicity Pattern Analysis

I identified platform-specific toxicity taxonomies that explain detection challenges.

4.4.1 Reddit Patterns. Reddit toxicity centers on personal capability and emotional discouragement (Table 6).

These target personal identity, emotional vulnerability, and self-efficacy.

4.4.2 GitHub Patterns. GitHub toxicity is technical, terse, and authority-driven (Table 7).

These focus on technical legitimacy, issue triage, and professional norms.

5 DISCUSSION

My findings reveal that toxicity detection effectiveness is shaped not by model sophistication alone but by alignment between model assumptions, platform norms, and linguistic culture. I discuss three central themes: platform-dependent performance, cultural-linguistic factors shaping toxicity, and design implications for automated moderation.

5.1 The Platform Dependency Paradox

The clearest result is the reversal in Detoxify's performance across platforms. Detection performance hinges on whether model framing and training match platform-specific linguistic norms.

Table 5: Genuinely Toxic Reddit Comments

Type	Example	Score
Dismissive	"talk to a therapist or something dude"	8
Gatekeeping	"you're not meant to be a coder"	8
Hostile	"you're a giant piece of shit"	10

5.1.1 Why Llama Succeeds. Llama performs well for three reasons. First, the prompt explicitly defines technical-community toxicity, providing a suitable task frame. Second, it encodes expectations for "helpful, beginner-friendly tone", enabling the model to treat terse responses as violations even when linguistically neutral. Third, Llama's semantic reasoning captures contextual insufficiency: comments like "won't fix," "user error," or "this is documented" become toxic when delivered without explanation. These patterns, free of hostile keywords are invisible to Detoxify.

5.1.2 Why Detoxify Fails on Reddit. Detoxify suffers from severe under-flagging on GitHub. This follows directly from its training: profanity and aggression in social media data become primary toxicity cues. Reddit's supportive profanity triggers near-universal high scores, while GitHub's professional terseness, despite being harmful in context, contains none of Detoxify's learned lexical markers. This demonstrates the limits of keyword-centric detectors when language is used differently across communities.

5.2 Cultural-Linguistic Factors in Toxicity

Qualitative analysis shows that what counts as toxic—and how toxicity is expressed—is shaped by platform norms rather than universal linguistic signals.

5.2.1 Reddit's Casual Supportive Culture. r/learnprogramming employs profanity as solidarity, emphasis, and emotional validation. Profane expressions mark encouragement ("debugging is fucking hard"), excitement ("holy shit I get it"), or shared struggle, not hostility. Within this culture, toxicity manifests not through aggression but through capability denigration, discouragement. These often appear polite or conversational, complicating detection. Models trained on general hostility signals over-detect supportive profanity and under-detect polite discouragement.

5.2.2 GitHub's Professional Efficiency Culture. GitHub emphasizes efficiency, conciseness, and technical precision. Explicit hostility is rare; instead, toxicity emerges through dismissive brevity. Comments like "duplicate," "not a bug," or "read the docs" can be technically correct yet unhelpfully terse, especially without links or rationale. For newcomers lacking project context, such responses feel exclusionary despite neutral wording.

Here, harmful behavior is defined by lack of context, not linguistic aggression, requiring detectors to reason about sufficiency rather than surface features.

5.3 Implications for Detection System Design

These findings challenge assumptions behind current moderation systems and highlight the need for context-aware, culturally aligned detection strategies.

5.3.1 No Universal Detector Exists. Both Detoxify and Llama demonstrate that models trained or prompted generically cannot handle specialized community norms. Detoxify fails because its social-media training data is mismatched; Llama fails when its GitHub-optimized prompt is applied to Reddit. This indicates that platforms cannot rely on off-the-shelf toxicity APIs without local validation. Community-specific tuning is a requirement, not an enhancement.

5.3.2 Context-Aware Detection Must Encode Cultural Norms. Effective toxicity detection requires explicit encoding of platform norms. For LLMs, this means prompts must specify culture-dependent signals:

- For GitHub: dismissiveness includes "technically correct but insufficient" responses.
- For Reddit: capability discouragement is toxic even when tone is soft; supportive profanity is acceptable.

For trainable models, curated platform-specific datasets are essential to capture distinct toxicity mechanisms. Cross-platform generalization cannot be assumed.

5.3.3 Qualitative Analysis Is Essential. Quantitative metrics alone obscure the reasons behind model errors. The qualitative inspection revealed that Detoxify relies on profanity frequency, GitHub toxicity often stems from terse insufficiency rather than hostility. Such insights are critical for designing better prompts, training data, and decision rules. Iterative, qualitative error analysis must accompany quantitative evaluation.

5.4 Limitations

This project has several limitations. **Ground truth:** Reddit lacks full labels; my estimate of ~25 toxic comments is based on reviewing only high-scoring cases. **Sample size:** The Reddit toxic sample is small; stronger conclusions require larger annotation efforts. **Platform specificity:** r/learnprogramming and GitHub repository sample may not reflect other subcommunities with different norms. **Temporal scope:** Data reflect 2023–2024 norms; toxicity patterns evolve over time. **Cultural scope:** English-language communities dominate the dataset; norms vary across cultures. **Method scope:** Only two methods were evaluated; fine-tuned, ensemble, or hybrid models may behave differently. **Precision:** Precision cannot be measured due to incomplete ground truth, limiting our ability to study false positives. **User impact:** We measure toxicity presence but not its impact on learning outcomes or participation decisions.

6 CONCLUSION

This project presents a systematic cross-platform examination of toxicity detection in programming support communities, highlighting fundamental challenges in applying automated moderation tools to technical contexts. Our findings show that toxicity is not a universal linguistic construct but a platform-shaped phenomenon that requires context-aware detection approaches.

6.1 Summary of Key Findings

Platform-Dependent Detection Performance: Detection effectiveness varies dramatically across platforms. This contrast demonstrates that even advanced LLMs do not generalize across platforms without context-specific optimization.

Table 6: Reddit Toxicity Pattern Taxonomy

Pattern	Characteristics	Examples
Social Dismissiveness	Invalidates help-seeking	"why post shit like this"
Capability Gatekeeping	Claims programming "not for you"	"if you suck at maths you're not meant to code"
Condescending Teaching	Advice framed as incompetence	"you suck at it because..."
Discouragement	Advises abandoning programming	"maybe software dev isn't for you"

Table 7: GitHub Toxicity Pattern Taxonomy

Pattern	Characteristics	Examples
Vulgarity	Making vulgar remarks	"Why is it so hard to install this thing? It has been 3 f**ing hours already!!!"
Insulting	Making insulting comments towards other participants	"It's really very annoying to debug [...] Is anybody alive here? Do you need reproduction? Or what?"
Unprofessional	Using unprofessional words	"Darn it, now I can't reproduce after a restart of the window manager."
Mocking	Mocking other users/tools	"This isn't Windows compatible at all, are you joking?"
Profane Technology Naming	Using unprofessional names	"plugins=(git.... thefuck.....)"
Threat	Making threatening remarks	"I think you're just trolling, so enjoy the ban and learn some manners."

Limitations of General-Purpose Models: Detoxify, trained on social media data, shows opposite failures on each platform. It misses most GitHub toxicity because professional dismissiveness contains no explicit abusive keywords. Conversely, it over-flags Reddit comments because supportive profanity appears toxic within its social-media-centric training distribution. These patterns indicate that models built for generic social environments cannot be directly applied to technical communities.

Platform-Specific Toxicity Patterns: Toxicity manifests differently across communities. Reddit toxicity often targets personal capability or identity (e.g., "maybe programming isn't for you"), aligning with its social learning context where emotional vulnerability is high. GitHub toxicity emerges through professional dismissiveness, reflecting efficiency-focused norms.

Role of Cultural-Linguistic Norms: Reddit's casual, supportive culture normalizes profanity used for encouragement or solidarity, which confuses keyword-based systems. GitHub's professional norms encourage terse responses that, while polite in form, can still be dismissive or exclusionary. Effective detection must therefore consider cultural expectations: profanity may be supportive, and professional tone may still mask harmful dismissiveness.

6.2 Final Remarks

Programming communities serve as essential spaces for learning and collaboration. Toxic interactions can discourage newcomers, hinder open-source participation, and undermine inclusivity. This project demonstrates that toxicity in technical communities is deeply shaped by platform culture, and automated detection systems must be tailored to these contexts to be effective.

A single detection method cannot perform reliably across platforms: the same LLM that achieves strong results on GitHub fails entirely on Reddit, and general-purpose models trained on social media struggle with both. Addressing toxicity in programming communities therefore requires a combination of technical solutions, cultural understanding, and qualitative analysis.

As online programming participation continues to grow, developing context-aware, platform-specific moderation strategies will be essential for maintaining healthy, welcoming communities.

REFERENCES

[1] Mia Mohammad Imran and Jaydeb Sarker. 2025. *"Silent Is Not Actually Silent": An Investigation of Toxicity on Bug Report Discussion.* Association for Computing Machinery, New York, NY, USA, 576–580. https://doi.org/10.1145/3696630.3728502