

Toxicity Ahead: Forecasting Conversational Derailment on GitHub

Mia Mohammad Imran, Robert Zita, Rahat Rizvi Rahman, Preetha Chatterjee, Kostadin Damevski
Missouri University of S&T / Elmhurst University / Virginia Commonwealth University / Drexel University

ICSE2026

背景：OSSにおける有害な発言

OSSコミュニティでは有害な（toxic）やり取りが問題になっている

- 2024年のGitHub調査：開発者の64.23%が否定的なやり取りを経験・目撃21.4%が、そのやり取りを理由に貢献をやめた

既存の有害発言の検出はすべて**事後的**（post-hoc）

- 有害な発言が投稿された後にしか見つけられず、被害を防げない

人手による会話の監視は負担が大きい

- issue, PR, 議論板など複数のチャネルを常に監視する必要がある

背景：GitHub上での有害な会話の見つけにくさ

GitHubの有害さは、X や Reddit のような露骨な攻撃とは異なる

- 権利の主張，誤解，新しいやり方への抵抗など，**微妙な形**で現れる

ソフトウェア開発特有の語が，一般的な有害検出ツールを誤らせる

- 例：kill, dead, dump などは技術用語として普通に使われる

➤ 有害になる前に気付くのは難しい

GitHub特有の難しさの中で、有害な発言が現れる前に気付く手がかりはないか？

背景：脱線を見つけるという発想

用語の定義

会話の脱線 (derailment)

建設的な議論が、否定的な方向に転じていく過程

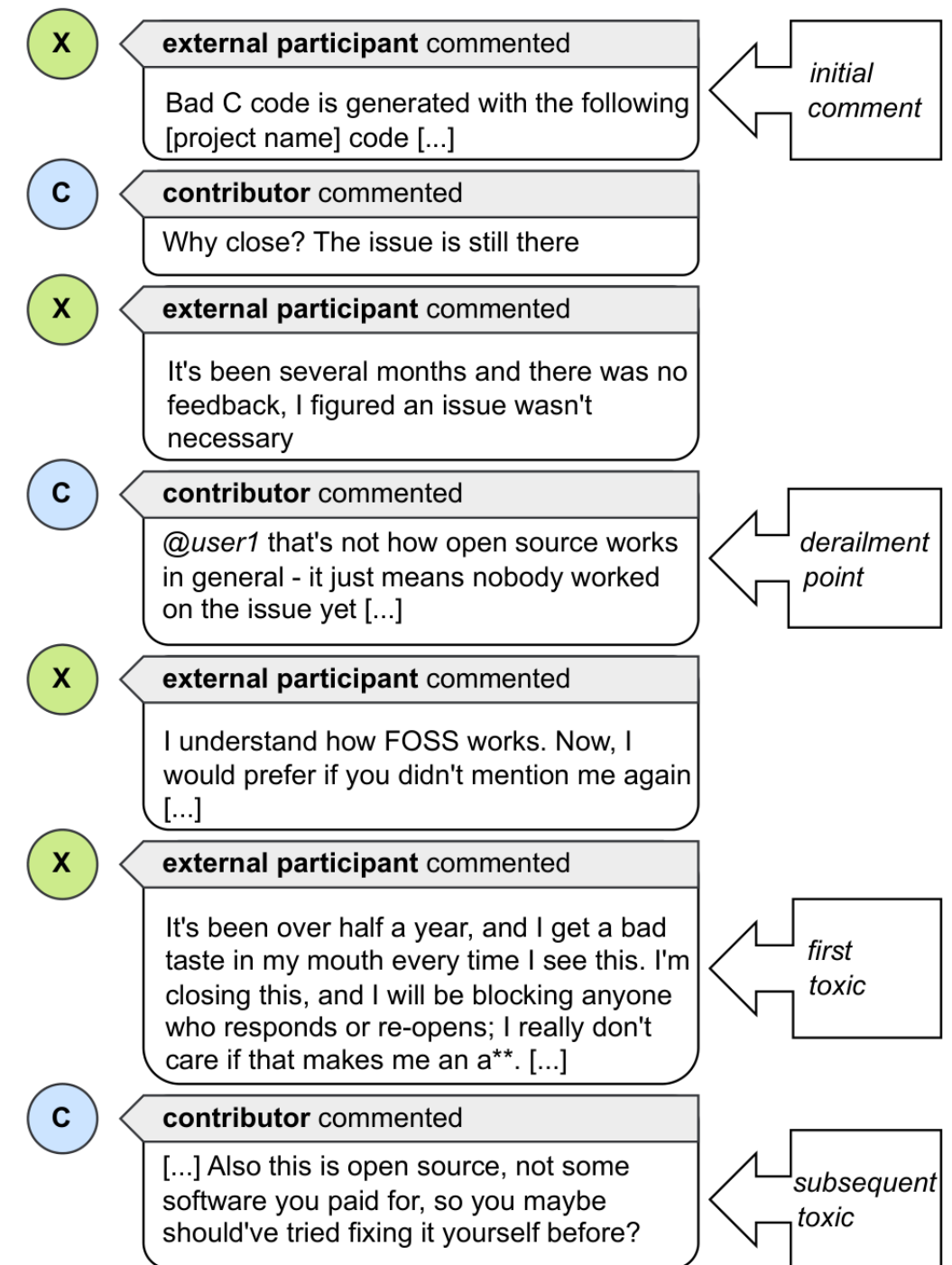
脱線点 (derailment point)

転換が起きたコメント。まだ有害ではない

会話の具体例(右図)：

- 脱線点：「オープンソースとはそういうもの。誰もまだ手を付けていないだけだ」
- 有害コメント：「反応する奴は全員ブロックする。最低な奴だと思われても構わない」

発想：脱線点以降に見られる兆候を手がかりに、有害発言が出る前に予測する



図：GitHub上の有害な会話の例（論文の Fig. 1）

研究概要

目的：

- GitHubで会話が有害な方向に脱線する特徴を分析し，有害発言が現れる前に脱線を予測する手法を提案する

リサーチクエスチョン：

- RQ1：GitHubの議論で，会話の脱線にはどのような特徴やパターンがあるか？
- RQ2：LLMは会話の脱線を効果的に予測できるか？
- RQ3：提案手法は独立したGitHubのデータにも一般化できるか？

データセット

脱線した有害スレッドデータセット構築

- Ehsani et al.のロックされた404スレッド（理由：過熱，スパム，話題外）から，GPT-4oと人手で有害発言を特定（ $\kappa = 0.78$ ）
- 2名で脱線点を注釈（ $\kappa = 0.914$ ）．突然有害になった34件などは除外

非有害スレッド：各有害スレッドの前後15件から収集し，人手で確認

データセット	内容	件数
脱線した有害スレッド	有害コメントの前に脱線点があるスレッド	159件（脱線点 382）
非有害スレッド	同じリポジトリの通常の issue / PR	207件

RQ1:脱線のタイミング

脱線した有害スレッド159件で、脱線点と最初の有害コメントの位置・時間を分析



脱線点から最初の有害コメントまでの時間

経過時間	割合	件数
8時間以内	46%	73 / 159
24時間以内	64%	102 / 159
7日以上後	25%超	40 / 159

脱線すると短時間で有害な発言に至りやすい → **早期の介入**が重要
ただし、長い時間をおいて有害化するケースもある

RQ1:脱線点の言語的特徴

表：各特徴を含むコメントの割合

脱線点は通常コメントより，二人称，否定，WH疑問等，
理由付けを伴う対立的な言い回しが多い

言語的特徴（例）	脱線点	有害	通常	Cramer's V
二人称代名詞（you, your）	60.7%	75.5%	43.9%	0.127
WH疑問詞（what, why, how）	57.1%	59.7%	43.9%	0.104
否定語（not, no）	70.2%	71.1%	55.3%	0.132
理由付け（because, since）	70.4%	70.4%	61.4%	0.055
強調（actually, really）	53.4%	59.7%	42.5%	0.123
伝達動詞（say, tell）	33.5%	36.5%	24.9%	0.133

RQ1:脱線点の口調ときっかけ

無礼な口調（TBDF）の割合

カテゴリ	脱線点（362）	有害コメント（142）
苛立ち（Bitter Frustration）	42.82%	24.65%
焦り（Impatience）	22.65%	9.15%
嘲り（Mocking）	9.94%	11.97%
侮辱（Insulting）	5.80%	25.35%

脱線のきっかけ（上位3つ，2名で注釈， $\kappa = 0.84$ ）

きっかけ	割合（159スレッド）
ツールやコードの失敗，エラーメッセージ	23.27%（37）
技術的な意見の対立	20.12%（32）
コミュニケーションの行き違い（誤解，誤字，言語の壁）	16.98%（27）

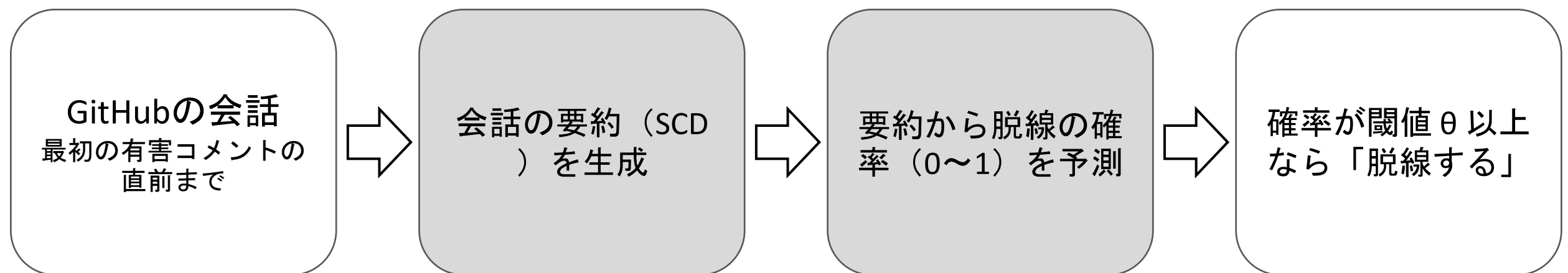
脱線点では苛立ちや焦りが多く，侮辱は有害コメントが多い
→ 微妙な不満が露骨な有害発言へと進む

RQ2:提案手法の概要

SCD (Summaries of Conversation Dynamics)

- 技術的な内容を除き, やり取りのパターン・感情・口調の変化を要約する

検出ではなく予測: **最初の有害コメントより前のコメントのみ**を入力



要約を挟むことで, **予測の根拠をモデレーターが読める** (説明可能性)
推論 (要約) と分類 (確率の予測) を分け, LLMの負担を減らす

RQ2:Least-to-Mostプロンプト

表：SCD生成プロンプトの7ステップ（簡単な観察から段階的に要約）

Step	内容	RQ1の知見との対応
1	会話の主要な要素（話題）を特定	－
2	除外基準：技術的な主張，コード名，引用を含めない	－
3	各参加者の意図（II：Individual Intentions）	Hua et al. の枠組み
4	会話の戦略（CF：Conversational Features）	言語的特徴
5	感情・口調の推移（STF：Sentiment and Tonal Features）	無礼な口調（TBDF）
6	緊張のトリガー（TT：Tension Triggers）	脱線のきっかけ
7	1～6を統合し，会話の軌跡を要約	－

※プロンプトの全文は付録を参照

RQ2:実験設定

LLM : Qwen2.5-32B-Instruct, Llama-3.3-70B

- temperature:0, コンテキスト長 32k
- GPT-4 や Claude などの有償モデルは使わない
 - 再現性とコストの観点

比較手法

- CRAFT : 脱線予測の代表的なニューラルネットワークモデル
- Hua et al. : 人手で書いたSCDの例を与える few-shot プロンプト

データ : 脱線した有害スレッド 159件 + 非有害スレッド 207件

評価 : 閾値 θ ごとに Precision, Recall, F1

- $\theta \in \{0.1, 0.3, 0.5, 0.7\}$

RQ2:予測の結果 (F1)

表：閾値 θ ごとの脱線予測の F1

Least-to-Most SCD は $\theta = 0.3$ で最高の F1 (Qwen **0.901**, Llama **0.852**)
CRAFT の最高 F1 は 0.580. ただし Llama の $\theta = 0.1$ のみ Hua et al. が上回る

モデル	手法	$\theta = 0.1$	$\theta = 0.3$	$\theta = 0.5$	$\theta = 0.7$
CRAFT	—	0.579	0.580	0.551	0.476
Qwen2.5-32B	Hua et al. (few-shot SCD)	0.612	0.747	0.410	0.281
	Least-to-Most SCD	0.614	0.901	0.644	0.491
Llama-3.3-70B	Hua et al. (few-shot SCD)	0.850	0.807	0.708	0.525
	Least-to-Most SCD	0.812	0.852	0.790	0.653

RQ2: 予測の結果 ($\theta = 0.3$)

表 : $\theta = 0.3$ での Precision, Recall, F1

Hua et al. は Precision は高いが Recall が低い（見逃しが多い）
Least-to-Most は Precision と Recall のバランスが良い

モデル	手法	Precision	Recall	F1
CRAFT	—	0.425	0.912	0.580
Qwen2.5-32B	Hua et al. (few-shot SCD)	0.980	0.604	0.747
	Least-to-Most SCD	0.945	0.862	0.901
Llama-3.3-70B	Hua et al. (few-shot SCD)	0.913	0.723	0.807
	Least-to-Most SCD	0.890	0.818	0.852

RQ2: アブレーション

表：プロンプトの要素を1つ除いたときの性能（Qwen, $\theta = 0.3$ ）

感情の推移（STF）と緊張のトリガー（TT）を除くとF1が有意に低下
主に Recall が下がる → 微妙な兆候を捉えるのに重要

除いた要素	Precision	Recall	F1	p値
個々の意図（II）	0.962（+1.8%）	0.792（-8.1%）	0.869（-3.6%）	0.2153
会話の戦略（CF）	0.941（-0.4%）	0.805（-6.6%）	0.868（-3.7%）	0.1628
感情・口調の推移（STF）	0.898（-5.0%）	0.774（-10.2%）	0.831（-7.8%）	0.0032
緊張のトリガー（TT）	0.906（-4.1%）	0.786（-8.8%）	0.842（-6.5%）	0.0190
なし（Full LtM）	0.945	0.861	0.901	—

RQ2:誤り分析

表：誤りの主な原因（Qwen, $\theta = 0.3$ ）

- 主な原因のみ抜粋（1件が複数に分類される場合あり）

誤りの種類	主な原因	件数
偽陽性（8件）	穏やかなやり取りで緊張を過大評価	3
	要約は正しいが、予測で深刻さを誤判定	3
偽陰性（22件）	苛立ちなど微妙な有害の兆候を見逃す	10
	皮肉やニュアンスのある口調を検出できない	3
	要約は正しいが、予測で深刻さを誤判定	3
	脱線から長時間経ってから有害化	3

口調の解釈（皮肉、微妙な苛立ち）と、
要約から確率の予測への受け渡しに弱点がある

RQ3:独立したデータでの評価

表：Raman et al. のデータでの F1（有害 65，非有害 243）

Least-to-Most の最高 F1：Qwen **0.797**（ $\theta = 0.3$ ），Llama **0.776**（ $\theta = 0.5$ ）
ただし $\theta = 0.1$ では Hua et al. を下回り，閾値の調整が重要

モデル	手法	$\theta = 0.1$	$\theta = 0.3$	$\theta = 0.5$	$\theta = 0.7$
Qwen2.5-32B	Hua et al.（few-shot SCD）	0.438	0.754	0.577	0.483
	Least-to-Most SCD	0.382	0.797	0.702	0.516
Llama-3.3-70B	Hua et al.（few-shot SCD）	0.708	0.734	0.707	0.586
	Least-to-Most SCD	0.656	0.735	0.776	0.744

考察：実践への示唆

GitHubのメンテナー向け

- LLMによる早期警告を、既存のモデレーションに組み込む
- 閾値で介入を変える： $\theta > 0.7$ はモデレーターに通知， $0.3 \sim 0.7$ はボットが穏やかに注意喚起

研究者向け

- プロンプトの感度を下げ、複数のプラットフォーム（BugZilla 等）で使えるベンチマークを作る
- コメントごとに要約を更新し、計算コストを下げる

結論

研究のアプローチ

- GitHubで脱線した有害スレッド159件と非有害スレッド207件を構築し、脱線の特徴を分析

発見1：脱線には予測可能な兆候がある

- 有害発言の直前（中央値3コメント前）に、二人称・否定・疑問、苛立ちや焦りが増える

発見2：Least-to-Most SCD で脱線を予測できる

- F1 0.901（Qwen）， 0.852（Llama）で既存手法を上回り，独立データでも最大0.797

要約を介した説明可能な予測で，事後対応から事前のモデレーションへ

所感

- あくまで脱線してから、その後実際に毒性コメントが発生しないかどうか予測する研究
- この研究の本質は提案手法ではなく、課題への対処法の着眼点を変えたこと
- 面白いが、これを常に動かすとなると、実用コストは結局高い

補足スライド

妥当性への脅威

構成概念妥当性

- すべての有害発言に脱線の兆候があるわけではない（突然の毒性は予測できない）
- 脱線点の注釈は人の判断に依存し，LLMの要約にも幻覚の可能性がある

内的妥当性

- LLMの出力はプロンプトや入力の小さな変化に影響される

外的妥当性

- GitHubのみが対象で，データセットも小規模
- 公開されたGitHubの議論がLLMの学習データに含まれ，性能が高く出ている可能性

(付録)Least-to-Mostプロンプト

Least-to-Most (LtM) SCD Generator Prompt

You are a skilled Conversation Analyst specializing in GitHub discussions. Your objective is to capture the conversation dynamics without getting caught in the technical details.

Your Analysis Method: Follow these steps in order, building your understanding from basic patterns to complex dynamics:

Step 1: Identify Main Elements: Identify the main elements by quickly scanning the conversation and pinpointing the key components or topics being discussed.

Step 2: Enforce Exclusion Criteria

Do NOT include:

- Any technical claims, arguments, or explanations
- Any code names, module names, or PR details
- Any direct or indirect quotations
- Any mention of what was being implemented or reviewed

Example 1 (Excluded): "The discussion addressed discrepancies in `evaluate_admissions()` outputs [...]."

Step 3: Note Individual Intentions

Infer what each participant is aiming to achieve. Example: [...]

Step 4: Identify Conversation Strategies

Identify rhetorical or structural tactics used by each speaker. Example: [...]

Step 5: Track Sentiment Evolution

Describe how the tone shifts across the conversation. Use

descriptors such as neutral, guarded, dismissive, confrontational, collaborative, defensive, escalating. Avoid inference. [...]

Step 6: Find Tension Triggers

Detect explicit shifts in tone, rhythm, or interaction style that signal rising tension—only when clearly supported by observable evidence. Avoid inference. [...]

Step 7: Synthesize Trajectory Summary

Combine insights from steps 1-6 to create a concise trajectory summary.

Remember: Your value lies in revealing the human dynamics beneath technical discussions. Focus on HOW people communicate, not WHAT technical points they make.

Example: "Multiple users debate reverting a recent PR.[...]"

Your Task: Using the above guideline, write the final trajectory summary for the following GitHub discussion transcript:

< *insert conversation transcript* >

Write only the final summary within double quotation.

(補足)TBDFとは

TBDF (Tone-Bearing Discussion Features) : 無礼な口調の分類

- Ehsani et al.[16] が定義. 本研究では脱線点の注釈と, 脱線点／有害コメントの口調の比較に使う (太字は本編に掲載の4カテゴリ)

カテゴリ	定義	例 (論文 Table 1 より)
苛立ち (Bitter Frustration)	強い不満, 不快感, 苛立ちの表明	No answer, no reaction, what kind of support is that.
焦り (Impatience)	遅れに対する不満の表明	Issue not fixed in 30 days? Must be gone!
嘲り (Mocking)	相手や物事を無礼な形でからかう	Legend says this will still exist even on the end of mankind.
皮肉 (Irony)	字義と逆の意味を持たせる言い回し	Maybe you should actually write that down somewhere.
下品 (Vulgarity)	攻撃的・不適切な言葉遣い	Who cares, same sh*t.
脅迫 (Threat)	否定的な結果をほのめかす警告	Any further responses will result in you being blocked [...]
権利の主張 (Entitlement)	特別扱いや特権を期待する	that's how good we are. I don't want your contribution. [...]
侮辱 (Insulting)	他者やプロジェクトへの中傷	This looks like it was done by a 5 year old.
属性攻撃 (Identity attacks)	人種, 宗教, 性別, 国籍等に基づく中傷	I would not be surprised if this database is maintained by the [nationality].