Contextualizing Toxicity in Open Source: A Qualitative Study

Sophie Cohen
Wesleyan University
Middletown, Connecticut, USA
scohen02@wesleyan.edu

ABSTRACT

In this paper, we study toxic online interactions in issue discussions of open-source communities. Our goal is to qualitatively understand how toxicity impacts an open-source community like GitHub. We are driven by users complaining about toxicity, which leads to burnout and disengagement from the site. We collect a substantial sample of toxic interactions and qualitatively analyze their characteristics to ground future discussions and intervention design.

CCS CONCEPTS

• **Software and its engineering → Open source model**.

KEYWORDS

open source, toxicity, classifier, sentiment analysis

ACM Reference Format:
Sophie Cohen. 2021. Contextualizing Toxicity in Open Source: A Qualitative Study. In *Proceedings of the 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '21), August 23–28, 2021, Athens, Greece.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3468264.3473492

1 INTRODUCTION

Toxicity encompasses a wide range of negative behaviors, including overt insults, rude and disrespectful comments, sarcasm, and microaggressions [2, 14]. Toxic interactions have been studied in many settings, specifically on social media [8], often with negative impacts on the well-being and continued engagement of participants exposed to such interactions.

Toxic behavior is increasingly reported and discussed in open-source communities [e.g., 1, 4, 5, 11, 20] and may be one important factor that makes these communities less welcoming, diverse, and sustainable [15]. In a space where, despite increasing professionalization and commercial influence, still the majority of participants volunteer their time [21], reports abound from open-source maintainers about stress, burnout or even complete disengagement due to toxic interactions [1, 4, 11, 15, 20]. Maintainers complain about these interactions, specifically those with aggressive or entitled undertones, such as in Figure 1. Attention to this topic is reflected in the increasing adoption of 'code of conduct' policies as a possible intervention [16, 18]. It has also been shown that different online communities have varying degrees of profanity, including the frequency, forms, and contexts, and thus detection systems must be adjusted for specific communities [17].

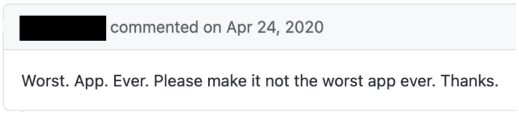

Figure 1: Excerpt from an issue discussion on GitHub.

Sentiment analysis for software engineering is a technique for analyzing issue discussions, pull requests, and forum posts [e.g., 3, 9]. There have been studies on detecting toxicity in language, such as hate speech, abuse, microaggressions, and harassment [6, 19]. Research has shown, however, that there are limitations of current sentiment analysis tools [12].

Specifically, in order to better understand the context of toxicity in open source, our research is guided by the following questions: (1) What forms of toxic interactions take place in open-source issue discussions? (2) When and how does toxicity occur? (3) What triggers toxicity? (4) Who are the people involved in toxic interactions and how do maintainers react?

Toxicity has been studied on other platforms, such as Twitter and Wikipedia [8, 10, 14], but we set out to see if it is different on GitHub. By understanding the state of toxicity on GitHub, we will be able to understand interventions in open source better and be able to help other communities.

2 APPROACH

Overall, we began our research by exploring the literature on toxicity in social and professional interactions. We first loosely defined toxicity according to the Google Perspective API, which defines toxicity as "a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion" [7]. However, realizing that characteristics of open-source work are different from traditional social and professional interactions, we decided that we need to ground our analysis in fresh unbiased data from open-source projects. We proceeded in two steps, as illustrated in Figure 2:

- First, we curated a dataset of toxic interactions in open source. We looked at four groups to identify potential toxic interactions: issues that were deleted, locked as too heated, mentioned the Code of Conduct, and that a classifier [15], which used multiple heuristics, marked as toxic. Subsequently, we manually *labeled* them as toxic or not toxic. These toxic interactions come from many different projects on GitHub.
- Second, we qualitatively analyzed toxic interactions and their context to identify patterns and trends in the data. Among others, we *coded* different forms of toxicity, people involved and their characteristics, as well as causes and reactions.

ESEC/FSE '21, August 23–28, 2021, Athens, Greece

ACM ISBN 978-1-4503-8562-6/21/08…$15.00
https://doi.org/10.1145/3468264.3473492

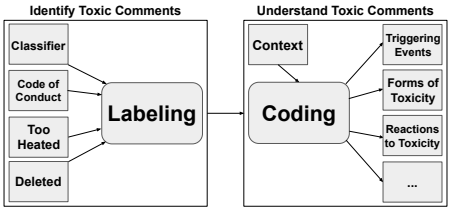

Figure 2: Overview of research methods

Category	Codes
Position of Toxic Comment	Opening with toxic comment (25), both opening with toxic comment and emerging from discussion (7), emerging from discussion (3)
What Triggered Toxicity	Failed use of tool/code (10), technical disagreement (9), politics/ideology (6), past interactions (4), N/A (4), unclear (2)
Target of Toxicity	Undirected (15), at people (11), at code (7), at company (3), self-directed (1)
Nature of Toxic Comment	Complaining (23), entitled (14), troll (8), joking (3), aggressive (2)
Severity of Toxic Comment	Cursing (15), colloquial (12), offensive (9), softer (7), unprofessional (3)
Who is the Author	Troll (9), repeated troll (8), experienced developer (7), project member (5), not active user (4), project owner (2), friend (1), deleted user (1)
Type of Project	Small (18), big active (13), dead/unimportant (6), corporate (5)
Domain of Project	App (8), games (6), end user (4), unclear (4), data science (3), web development (3) bitcoin (2), file sharing (2), political (2), library (1)
What Happens Afterwards	Closed (25), turning constructive (12), discussion (7), escalating further (3), no reaction (1)
Immediate Consequences (does not apply to most issues)	N/A (27), lots of effort for constructive reaction (4), emotional involvement visible (2), careful response/explaining (1), maintainer is affected (1)

Figure 3: Coding framework

This process allowed us to collect a random sample of confirmed toxic issues that we are proceeding to qualitatively code. We had a bottom-up coding strategy in which the codes were created organically. We utilized a mixed-methods study design by collecting issues from classifiers and then qualitatively coding them. We used what we learned from past literature and constructed a taxonomy composed of codes. This process went from theoretical to axial to initial codes [13]. We labeled the possible toxic comments that the classifier produced, and to ensure inter-rater reliability, four of us separately marked comments as toxic or non-toxic. The percentage agreement for issues with two labels was 88.7%, and that for issues with three labels was 83.3%.

We met as a group to discuss each confirmed toxic issue from GitHub. We examined the context of 35 toxic issues from 35 different projects, and then we wrote down general descriptions of the progression of each issue. After open coding, we built a coding frame with 10 categories and the codes within each category, as in Figure 3. We then proceeded to do axial coding with that coding frame. These codes were grounded in the data. We had two or three people meet as a group to assign the codes to issues together. Each toxic interaction is classified using all categories of the coding framework, and multiple codes within a category may be assigned to one issue. The number of issues that were classified as each code can be seen in parentheses in Figure 3. We are now continuing to analyze and assign these qualitative codes to more threads; our sample has 100 issue discussions with 40 from the classifier, 20 deleted, 20 locked, and 20 Code of Conduct.

3 PRELIMINARY RESULTS

Our results are based on the initial subset of 35 issues, as we are in the process of coding 65 more. The code frame helped us identify a large range of toxic behaviors. So far, the majority of users who write toxic comments we consider as trolls (17 of the 35 issues) because they have not opened an issue before and have essentially no other activity on GitHub. Repeated trolls were also very common (8 of the 17 trolls), which we defined as a user repeatedly opening an issue like the one at hand and have essentially no other activity on GitHub. Complaining was the most common form of toxicity (23/35), and this often came about with tones of entitlement (14/35). While a number of issues only used colloquial toxic language (12/35), many contained strong cursing (15/35). These toxic occurrences were most often undirected (15/35), but it was quite common for toxicity to be directed at another person (11/35). These toxic issues were usually closed (25/35), and many turned constructive, which means that the situation was diffused (12/35). We defined a small or big project based on the number of contributors, watches, and stars. In the sample, most of them were small projects (18/35).

We can compare our preliminary results to what has been found for Twitter and Wikipedia. There have been a number of analyses of harassment on Twitter, yet not much is known about the nature of such tweets or the people who write them [8]. After our analysis of GitHub toxic issues, we found a common theme of cursing and colloquial language. We hypothesize that profanities are also used on other domains, and therefore it is important to identify such words or phrases. We manually analyzed users involved in toxic instances, including the personal information on their profile, their activity, and how they have behaved on other projects. Our findings show that trolls are most commonly the authors of toxic comments; this could help predict who will be writing future toxic comments, and thus can suggest the need to monitor such users more carefully.

Moreover, it has been shown that context has a statistically significant effect on toxicity annotation for Wikipedia posts [14]. However, this study limited the notion of context to the previous post in the thread and the discussion title, while our study looks at the entire thread, the project as a whole, and the accounts of users involved. In a GitHub issue, users may be responding to comments that came well before the previous one. We believe that examining the entire context of the issue is crucial in understanding where toxicity comes from and what it leads to; these components will be useful for future tools to detect occurrences of toxicity.

4 CONTRIBUTIONS

We argue that toxicity in open-source communities comes in numerous forms and generally impacts users in a negative way. We suggest four groups (issues that were deleted, locked as too heated, mentioned the Code of Conduct, and that a classifier marked as toxic) as a way to identify possible toxic interactions in GitHub issue discussions, and provide a coding framework to qualitatively analyze confirmed toxic issues. We believe that toxicity can come in different forms on GitHub compared to that which has been studied on domains like Twitter or Wikipedia. Our results show preliminary statistics on the context of toxic occurrences – forms of toxicity, when and how it occurs, what triggers it, and who is involved. These findings can help to design better classifiers to detect toxic instances. The long-term vision of our research is to design and deploy effective interventions to mitigate toxicity and its effect in open-source communities, making them more welcoming and sustainable.

REFERENCES

[1] Heather Arthur. 2013. Being Ridiculed for My Open Source Project. https://harthur.wordpress.com/2013/01/24/771/ Blog post.

[2] Luke Breitfeller, Emily Ahn, David Jurgens, and Yulia Tsvetkov. 2019. Finding microaggressions in the wild: A case for locating elusive phenomena in social media posts. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 1664–1674.

[3] Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. 2018. Sentiment polarity detection for software development. *Empirical Software Engineering* 23, 3 (2018), 1352–1382.

[4] Brett Cannon, Adam Stacoviak, and Jerod Santo. 2018. The Changelog, Episode 318: A call for kindness in open source. https://www.changelog.com/podcast/318 Podcast.

[5] Kevin Daniel André Carillo and Josianne Marsan. 2016. "The Dose Makes the Poison"-Exploring the Toxicity Phenomenon in Online Communities. (2016).

[6] Darja Fišer, Ruihong Huang, Vinodkumar Prabhakaran, Rob Voigt, Zeerak Waseem, and Jacqueline Wernimont (Eds.). 2018. *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Association for Computational Linguistics, Brussels, Belgium. https://www.aclweb.org/anthology/W18-5100

[7] Google. [n.d.]. Perspective API. https://www.perspectiveapi.com/

[8] Joshua Guberman, Carol Schmitz, and Libby Hemphill. 2016. Quantifying toxicity and verbal violence on Twitter. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion*. 277–280.

[9] Emitza Guzman, David Azócar, and Yang Li. 2014. Sentiment analysis of commit comments in GitHub: an empirical study. In *Proceedings of the 11th working conference on mining software repositories*. 352–355.

[10] Mai Ibrahim, Marwan Torki, and Nagwa El-Makky. 2018. Imbalanced toxic comments classification using data augmentation and deep learning. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 875–878.

[11] Nolan Lawson. 2017. What it feels like to be an open-source maintainer. https://nolanlawson.com/2017/03/05/what-it-feels-like-to-be-an-open-source-maintainer/ Blog post.

[12] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, Michele Lanza, and Rocco Oliveto. 2018. Sentiment analysis for software engineering: How far can we go?. In *Proceedings of the 40th International Conference on Software Engineering*. 94–104.

[13] Matthew B Miles, A Michael Huberman, and Johnny Saldaña. 2018. *Qualitative data analysis: A methods sourcebook*. Sage publications.

[14] John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. 2020. Toxicity Detection: Does Context Really Matter? *arXiv preprint arXiv:2006.00998* (2020).

[15] Naveen Raman, Minxuan Cao, Yulia Tsvetkov, Christian Kästner, and Bogdan Vasilescu. 2020. Stress and Burnout in Open Source: Toward Finding, Understanding, and Mitigating Unhealthy Interactions. In *Proceedings of the Proc. International Conference on Software Engineering – New Ideas Track (ICSE-NIER)*.

[16] Vandana Singh and William Brandon. 2019. Open source software community inclusion initiatives to support women participation. In *IFIP International Conference on Open Source Systems*. Springer, 68–79.

[17] Sara Sood, Judd Antin, and Elizabeth Churchill. 2012. Profanity use in online communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1481–1490.

[18] Parastou Tourani, Bram Adams, and Alexander Serebrenik. 2017. Code of conduct in open source projects. In *2017 IEEE 24th international conference on software analysis, evolution and reengineering (SANER)*. IEEE, 24–33.

[19] Zeerak Waseem, Wendy Hui Kyong Chung, Dirk Hovy, and Joel Tetreault (Eds.). 2017. *Proceedings of the First Workshop on Abusive Language Online*. Association for Computational Linguistics, Vancouver, BC, Canada. https://doi.org/10.18653/v1/W17-30

[20] Tim Wood. 2016. moment().endOf('term'). https://medium.com/timrwood/moment-endof-term-522d8965689 Blog post.

[21] Frances Zlotnick. 2017. GitHub Open Source Survey 2017. http://opensourcesurvey.org/2017/ doi: 10.5281/zenodo.806811.