UNIVERSITY OF TORONTO
FACULTY OF INFORMATION

Sentiment within the newcomer Linux Kernel Development Community

Montgomery Gole & Dr. Kelly Lyons

NSERC
CRSNG

A Longitudinal Study on the Maintainers' Sentiment of a Large Scale Open Source Ecosystem

Isabella Ferreira
Polytechnique Montréal
isabella.ferreira@polymtl.ca

Kate Stewart
Linux Foundation
kstewart@linuxfoundation.org

Daniel German
University of Victoria
dmg@uvic.ca

Bram Adams
Polytechnique Montréal
bram.adams@polymtl.ca

***Abstract*—Software development is a collaborative activity in which feelings and emotions can affect the developer's productivity, creativity, and contribution satisfaction. For example, the Linux Kernel Mailing List (LKML), which is used by subsystem maintainers to review patches sent by contributors, is known for its direct communication style, which is sometimes blamed as having a negative impact on contributors. In September 28, 2018, the kernel's lead maintainer, Linus Torvalds, announced that he would take a temporary break from the community, which led numerous members of the kernel community and observers from other communities to wonder to what extent this unexpected event could raise awareness about respectful interactions between community members. This paper performs an exploratory study in which we use an off-the-shelf sentiment mining tool to assess whether the maintainers' sentiment changed after Linus Torvalds' temporary break from his maintainer role. Based on the data available thus far, we did not find any high-level changes in maintainer sentiment. In future work, we will perform more fine-grained sentiment analysis.**

I. Introduction

Software development is a collaborative activity where developers use mailing lists, software code repositories, and issue tracking tools to manage their work [1]. Thus, like any collaborative activity, software development involves sentiment or emotions that can affect the developer's productivity, creativity, and contribution satisfaction [1]. The discussions can get excited as different people from distinct cultures and interests are part of it. In this sense, the risk of offensive behaviors increases more and more [2]. As a consequence, modern open source communities establish a code of conduct, which are a selection of ground rules for communication between participants, with the goal to make everyone comfortable in contributing to the open source project [2].

In the Linux kernel, all development activities are organized across a large number of mailing lists centered around the Linux Kernel Mailing List (LKML). This mailing list acts as a central place for discussions about technical and organizational aspects of the kernel development [3]. Contributors submit patches to subsystem maintainers, who approve the submissions and initiate the process of integrating the patch into an upcoming release [4]. Linus Torvalds, the founder of the Linux operating system, has the final decision as a project maintainer.

On September 16, 2018, Linus announced in LKML that he would temporarily take a break from Linux maintainership: *"My flippant attacks in emails have been both unprofessional and uncalled for. Especially at times when I made it personal. In my quest for a better patch, this made sense to me. I know now this was not OK and I am truly sorry. I am going to take time off and get some assistance on how to understand people's emotions and respond appropriately"* [5]. After this, many members of the kernel community as well as observers from other communities are wondering to what extent this unexpected event could raise awareness about respectful interactions between community members. The topics of politeness, positive interactions, and openness to opinions have become more openly discussed.

Thus, this paper aims to study whether this debate has changed the way in which kernel maintainers interact with developers and with each other. For this, we empirically study the sentiment in two years of kernel review discussions. In particular, we analyzed 15,166 e-mail threads that contain at least one subsystem maintainer from January 2017 to January 2019 to answer the following research questions:

***RQ1*. How does the e-mail thread sentiment evolve in the kernel review discussions?**
We could not find any significant difference in the sentiment across releases, months, and weeks. However, there is a slight difference in sentiment when comparing weekdays. Sunday, Tuesday, Wednesday, and Thursday are the days with more positive e-mail threads compared to negative e-mail threads.

***RQ2*. Did the maintainers' sentiment change after Linus Torvalds' temporary break from the community?**
There is no significant difference in the sentiment before and after Linus took a temporary break when comparing releases. However, there is a slight difference when comparing months, and weeks. The sentiment is more positive around August and September, and in November it becomes more negative.

II. The Linux Kernel Review Process

The Linux kernel still uses low-tech reviewing environments based on mailing list discussions [6], [7]. The patch submission process for the kernel is managed by Linus Torvalds, who makes the final decision, yet is guided by an army of "maintainers" in charge of vetting patches to one or more kernel subsystems. Figure 1 presents the life cycle of a typical kernel patch.

Empir Software Eng (2018) 23:953–986
DOI 10.1007/s10664-017-9536-y

CrossMark

How the R community creates and curates knowledge: an extended study of stack overflow and mailing lists

Alexey Zagalsky[1] · Daniel M. German[1] · Margaret-Anne Storey[1] · Carlos Gómez Teshima[1] · Germán Poo-Caamaño[1]

Published online: 18 August 2017

Abstract One of the effects of social media's prevalence in software development is the many flourishing communities of practice where users share a common interest. These large communities use many different communication channels, but little is known about how they create, share, and curate knowledge using such channels. In this paper, we report a mixed methods study of how one community of practice, the R software development community, creates and curates knowledge associated with questions and answers (Q&A) in two of its main communication channels: the R tag in Stack Overflow and the R-Help mailing list. The results reveal that knowledge is created and curated in two main forms: participatory, where multiple users explicitly collaborate to build knowledge, and crowd-sourced, where individuals primarily work independently of each other. Moreover, we take a unique approach at slicing the data based on question score and participation activities over time. Our study reveals participation patterns, showing the existence of prolific contributors: users who are active across both channels and are responsible for a large proportion of the answers, serving as a bridge of knowledge. The key contributions of this paper are: a characterization of knowledge artifacts that are exchanged by this community of practice; the

Communicated by: Romain Robbes, Christian Bird, and Emily Hill

✉ Alexey Zagalsky
alexeyza@uvic.ca

Daniel M. German
dmg@uvic.ca

Margaret-Anne Storey
mstorey@uvic.ca

Carlos Gómez Teshima
cagomezt@uvic.ca

Germán Poo-Caamaño
gpoo@uvic.ca

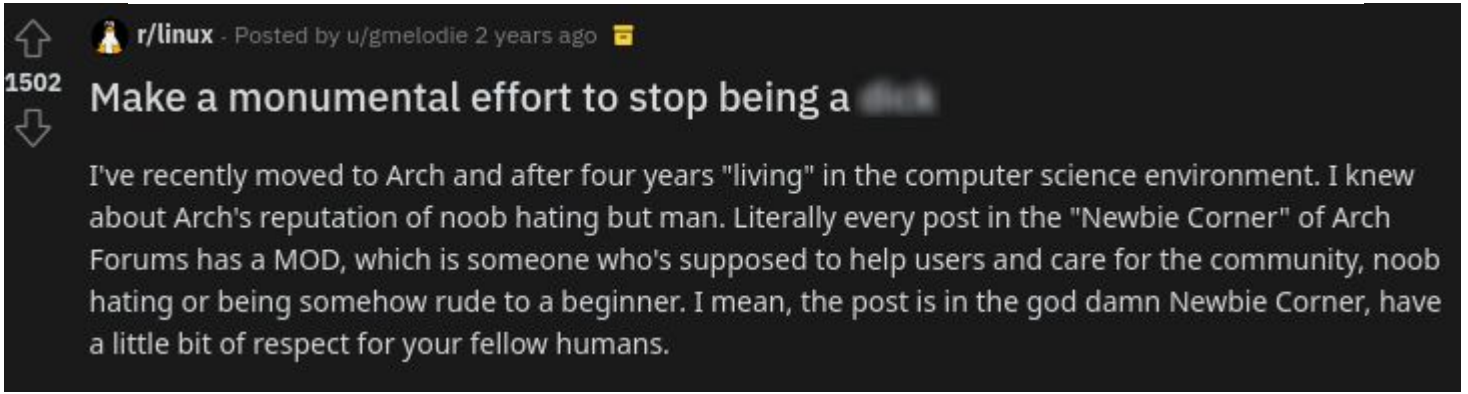
[1] University of Victoria, Victoria, BC, Canada

stackoverflow

Motivations

The "Shut the fk up" Phenomenon: Characterizing Incivility in Open Source Code Review Discussions**

ISABELLA FERREIRA, Polytechnique Montréal, Canada
JINGHUI CHENG, Polytechnique Montréal, Canada
BRAM ADAMS, Queen's University, Canada



Research Question

What is the sentiment like within newcomer spaces of the Linux Kernel Development community?

Data Collection

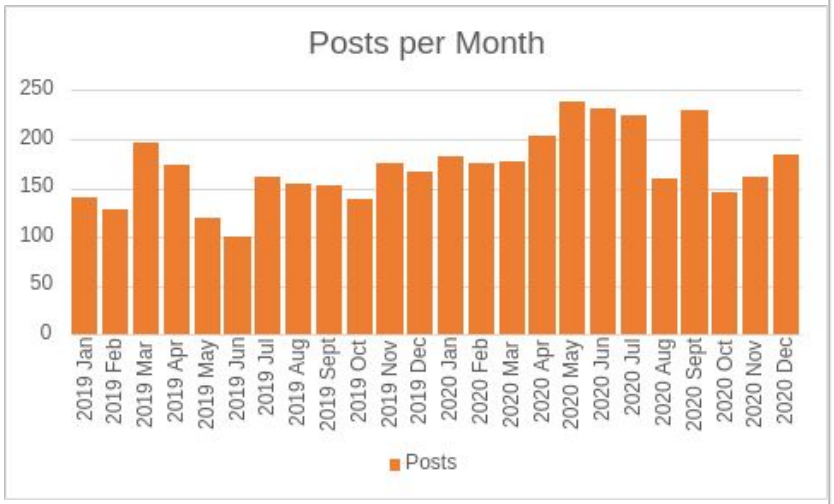
- Collection of data from both communities from January 2019 - December 31st 2020
- StackOverflow data extracted from database into CSV
  - Manipulated, understood through Microsoft Excel
  - Used RegEx to remove unwanted characters from Q&A text
- Mailing List data parsed with Python
  - Removed unwanted data from emails
  - Converted threads into JSON

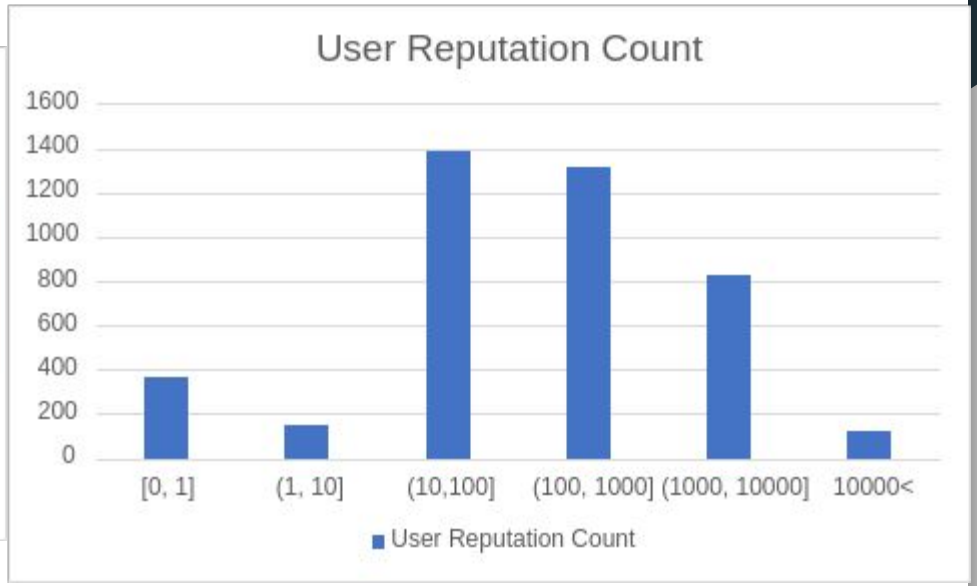
```
30      class Message:
31          def __init__(self, sender, datetime, subject, message_id, replying_to):
32              self.sender = sender_format
33              self.datetime = datetime
34              self.subject = subject
35              self.message_id = message_id
36              self.replying_to = replying_to
37      overall_format="From\s([a-z])+\sat\s([a-z])+.[a-z]+\s\s(Mon|Tue|Wed|Thu|Fri|Sat|Sun){1}\
38      sender_format = "From\s"
39      senderv2 = "From\s([a-z])+\sat\s([a-z])+.[a-z]+\s\s(Mon|Tue|Wed|Thu|Fri|Sat|Sun){1}\s(Ja
40      datetime = "Date:\s"
41      subject = "Subject:\s"
42      message_id = "Message-ID:\s"
43      reply = "In-Reply-To:\s"
44      message_format = "Message-ID:\s[^\n]+\n[\s\S]+?From\s([a-z])+\sat\s([a-z])+.[a-z]+\s\s(M
```

Database Schema	
Posts	
Id	int
PostTypeId	tinyint
AcceptedAnswerId	int
ParentId	int
CreationDate	datetime
DeletionDate	datetime
Score	int
ViewCount	int
Body	nvarchar (max)
OwnerUserId	int
OwnerDisplayName	nvarchar (40)
LastEditorUserId	int
LastEditorDisplayName	nvarchar (40)
LastEditDate	datetime
LastActivityDate	datetime
Title	nvarchar (250)
Tags	nvarchar (250)
AnswerCount	int
CommentCount	int
FavoriteCount	int
ClosedDate	datetime
CommunityOwnedDate	datetime
ContentLicense	varchar (12)

Users	
Id	int
Reputation	int
CreationDate	datetime
DisplayName	nvarchar (40)
LastAccessDate	datetime
WebsiteUrl	nvarchar (200)
Location	nvarchar (100)
AboutMe	nvarchar (max)
Views	int
UpVotes	int
DownVotes	int
ProfileImageUrl	nvarchar (200)
EmailHash	varchar (32)
AccountId	int

StackOverflow Database Schema


Posts per Month
250
200
150
100
50
0
2019 Jan
2019 Feb
2019 Mar
2019 Apr
2019 May
2019 Jun
2019 Jul
2019 Aug
2019 Sept
2019 Oct
2019 Nov
2019 Dec
2020 Jan
2020 Feb
2020 Mar
2020 Apr
2020 May
2020 Jun
2020 Jul
2020 Aug
2020 Sept
2020 Oct
2020 Nov
2020 Dec
Posts


User Reputation Count
1600
1400
1200
1000
800
600
400
200
0
[0, 1]
(1, 10]
(10,100]
(100, 1000]
(1000, 10000]
10000<
User Reputation Count

Data Processing

README.md

Senti4SD

Senti4SD is an emotion polarity classifier specifically trained to support sentiment analysis in developers' communication channels. Senti4SD is trained and evaluated on a gold standard of over 4K posts extracted from Stack Overflow. It is part of the Collab Emotion Mining Toolkit, (EMTk).

Fair Use Policy

Please, cite the following paper if you intend to use our tool for your own research:

> Calefato, F., Lanubile, F., Maiorano, F., Novielli N. (2018) "Sentiment Polarity Detection for Software Development," *Empirical Software Engineering*, 23(3), pp:1352-1382, doi: https://doi.org/10.1007/s10664-017-9546-9. (BibTeX)

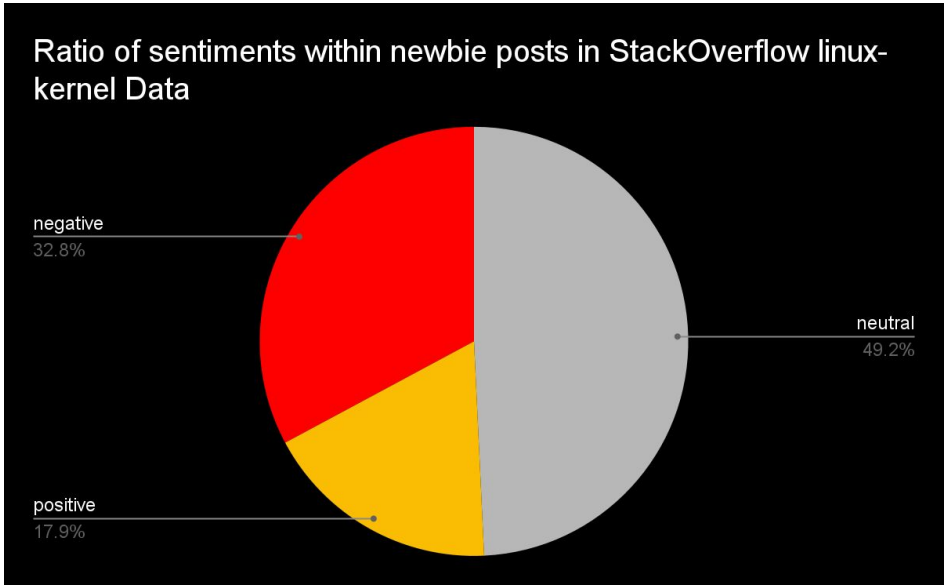
NOTE: You will need to install Git LFS extension to check out this project. Once installed and initialized, simply run:

```
$ git lfs clone https://github.com/collab-uniba/Senti4SD.git
```

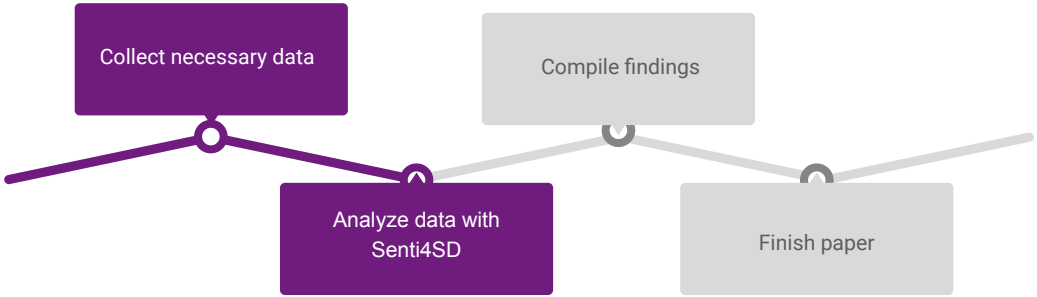
How do I get set up?

To set up the tool, simply run the following script from the command line:

```
$ sh requirements.sh
```



Next Steps


Collect necessary data
Analyze data with Senti4SD
Compile findings
Finish paper

Info About Me

Name: Montgomery Gole

Email: montygole@gmail.com

I’m applying to grad school for computer science!