BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Exploratory Analysis of Developer Sentiment On Open Source Projects

Md. Kawsar Ahamed Siam
190041202
Mahmudur Rahman
190041206
Moudud Hassan
190041240

Department of Computer Science and Engineering
Islamic University of Technology
June, 2024

Exploratory Analysis of Developer Sentiment On Open Source Projects

Md. Kawsar Ahamed Siam
190041202
Mahmudur Rahman
190041206
Moudud Hassan
190041240

Department of Computer Science and Engineering
Islamic University of Technology
June, 2024

Declaration of Candidate

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by **Md. Kawsar Ahamed Siam**, **Mahmudur Rahman**, and **Moudud Hassan** under the supervision of **Shohel Ahmed**, Assistant Professor, Department of Computer Science and Engineering, Islamic University of Technology, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

Shohel Ahmed
Assistant Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: June 04, 2024

Md. Kawsar Ahamed Siam
Student ID: 190041202
Date: June 04, 2024

Mahmudur Rahman
Student ID: 190041206
Date: June 04, 2024

Moudud Hassan
Student ID: 190041240
Date: June 04, 2024

Dedicated to our parents, whose support was integral to our work.

Contents

List of Figures

List of Tables

List of Abbreviations

OSS	Open Source Software
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations from Transformers
seBERT	Software Engineering Bidirectional Encoder Representations from Transformers
VCS	Version Control System

Acknowledgement

We are profoundly grateful to my supervisor, Shohel Ahmed, for his endless patience and insightful critiques. His expertise and encouragement were instrumental in the completion of this thesis. And to our parents, whose love and support have been a constant source of strength. Thank you for always being there.

Abstract

Issue-tracking platforms such as Jira and Bugzilla have become essential in large-scale software development. Prominent organizations in the Open Source Software (OSS) landscape, such as Apache and Mozilla, make heavy use of these platforms and document their software development process through online repositories that utilize version control systems (VCS) like Git. Artifacts gathered from these sources contain natural language data that can be used to answer important questions relating to the nature of the software produced and the sentiment of the developers. The commit frequency and working time of the developers can be correlated to the sentiment shown through the commit messages. Moreover, the sentiment of issue comments might differ significantly based on the type (i.e., bug or non-bug) or severity. In this regard, we utilized a modern machine learning-based approach through fine-tuning seBERT, a BERT model pre-trained on software development data, to classify sentiment and provide answers to these questions. We used an existing data set, 20-MAD, to test these hypotheses and provide the results. We found that high committer frequency is associated with a higher proportion of negative sentiments compared to low and medium frequencies, while the part of the day developers work in has minimal effect on measured sentiment. We also observed that the severity of an issue significantly influences the sentiment expressed in issue comments and issues classified as bugs have a higher negative sentiment frequency compared to other issue types combined.

Chapter 1

Introduction

In this chapter, we establish the underlying concepts, research significance, and a comprehensive overview of our work. The organization of this report has been elaborated on in the very last section of this chapter.

1.1 Overview

Sentiment Analysis is the computational study of people's opinions, attitudes, and emotions toward an entity [26]. Sentiment Analysis models human emotion and provides a framework for analyzing textual data of sentimental value. There are two primary psychological dimensions of sentiment: valence (positive or negative) and arousal (intensity) [36]. Some studies have also considered the emotion of the text, which can be classified into categories such as joy, sadness, anger, and fear [10]. However, in practice, a simpler model is used that classifies the text as positive, negative, or neutral. As such, it is a classification task: a piece of text may be regarded as positive, negative, or neutral sentiment, depending on the conditions set beforehand. These conditions vary depending on the domain of the data being analyzed and the questions being asked. Within the software development domain, several sources of textual data can be analyzed for sentiment such as commit messages, issue reports, and code comments, enabling us to make useful assumptions about the larger population of developers [11].

The flourishing of Open Source Software (OSS) in the last few decades has given birth to new opportunities for sentiment analysis for software engineering (SA4SE) [46] [2]. The software domain is unique in that it is a collaborative effort, and the emotional states of developers can significantly impact the quality of the software being developed. Sentiment analysis in the software domain can be used to understand the emotional states of developers, which can be used to determine the quality of the software being developed [15]. The sentiment gathered from software development artifacts can be used to study developer emotions, productivity, code quality, team dynamics, and more [30]. Using relational data such as sentiment class calculated from natural language, commit time, timezone, geographical distribution, project approval, status, severity, or type of bug-tracker issues, we can ask important questions about the software development landscape.

Commits are batches of changes that are sent to Version Control Systems (VCS) for tracking. Each commit in version control systems like Git or SVN are accompanied by a short, relevant message that describes the intent and contents of the commit. This message is usually in plain natural language and can therefore be used for sentiment analysis. Commit frequency, or regularity of a particular developer, on the other hand, can be an indication of their productivity. The frequency with which a developer commits can be related to the sentiment of the commit messages they write [42]. Alternatively, the working time of the developers might also play a role in this relation.

Issue tracking systems like Jira and Bugzilla have become the norm in large-scale software development. Issues can be classified into different types, namely, bug and non-bug [41]. In addition, issues are created with varying levels of severity to better model the things they represent: blocker, critical, major, minor, and trivia [43]. Issues contain discussions between developers relevant to the subject of the issue. The issue comments are of interest for their natural language textual data that enable sentiment analysis [25].

In this regard, we have determined several research questions not addressed in the existing literature relevant to commits and issue comments. Moreover, we use modern tooling based on statistical learning to extract sentiment as opposed to contemporary rule-based solutions. Finally, we make observations from the data and validate our findings through hypothesis testing.

1.2 Problem Statement

The frequency of commits, in conjunction with the sentiment of subsequent commit messages, can be used to determine the emotional implications of frequent committers for individual projects. Existing research explores this relation with aspects other than commit frequency, and as such there is an opportunity for exploring this dynamic. Developer working time is another parameter that can be used to ask questions about the sentimental state of the commit messages produced by developers who tend to work during particular parts of the day.

Another line of exploratory research in this domain takes the focus away from individual developers and studies them fro within the context of collaborative endeavors, e.g., data from bug-trackers such as Jira or Bugzilla. The sentiment shown on the comments of a particular issue might be related to the specific type of the issue, and given a large enough dataset of issues, significant differences can become visible between distinct issue types. The same can be said for the severity of the issues, conventional understanding of which is that more severe issues tend to attract more negative sentiment and vice versa.

From this reasoning, we have determined 4 Research Questions (RQs) in the domain of software development and maintenance for our study that can be answered by a sincere examination of software development artifacts.

- **RQ1:** Does Commit frequency affect the developer sentiment on the commit message?
- **RQ2:** Does working time affect the developer's sentiment? Is the relation reflected on commit messages?
- **RQ3:** Is the developer emotion in the issue comment dependent to issue type, e.g. Bug?
- **RQ4:** Do the emotions in the issue comments differ depending on the severity of the issue?

1.3 Research Challanges

In recent times, it has become harder to extract data from the likes of Github due to usage restrictions set on public APIs [5]. One might explore alternative APIs like GraphQL to circumvent these restrictions, but this requires a deep understanding of the API and the data it provides. This, paired with the fact that GraphQL APIs, while more capable in reducing overfetching compared to RESTful ones, are not allowed to get too complicated due to exponentially growing cloud compute costs, making more crude methods of data collection necessary. This often leads to another issue: large datasets such as 20-MAD [6] often include corrupted, incomplete, or unreliable data. This is due to the limitations of the mining potential of large software development data sources that account for complicated data cleaning steps that need to be architected to ensure that the data are reliable and accurate and that the integrity of the dependent research is maintained [19].

On the other hand, traditional sentiment analysis tools are not able to capture the context of the text and so cannot address negations and sarcasm [32]. Moreover, they are not able to handle new words and phrases that are not present in the dictionary, e.g., emoticons and emojis. Machine learning approaches, on the other hand, capture the context of the text and can therefore handle negations and sarcasm. They can also handle new words and phrases that are not present in the dictionary. In general, transformer architectures outperform rule-based approaches in sentiment analysis tasks. However, these models require a large amount of labeled data for training, which can be difficult to obtain, popularizing the use of transfer learning. This allows the model to leverage the knowledge learned from the first task to perform better on the second task [29] [28]. However, this is still largely a new field and so there is no standardized transformer-based tooling for sentiment analysis in the software development domain.

1.4 Contribution

We processed the unlabeled 20-MAD [6] dataset to form our working dataset. Then we fine-tuned seBERT [39] over 5 labeled datasets for the sentiment analysis task in the software engineering domain and calculated the sentiment classes of the git commit messages and issue comments for our working dataset. We then proposed a metric to calculate the commit frequency for developers and used the values to determine whether the commit frequency affects the developer's sentiment. Then we calculated the part of the day the commits were made and found the relation between working time and commit message sentiment. From there we determined if developer emotion in issue comments is dependent to the types of the issues being commented on. Likewise, we also determined if the severity of the issue has significant impact on the sentiment of the comments. We used hypothesis testing to validate and form our findings.

1.5 Organization

In chapter 2, we talk about related works that have influenced our study both directly and indirectly.

In chapter 3, we listed our whole methodology consisting of our dataset preparation, processing, and the proposed methodology for our research.

In chapter 4, we outlined and interpreted the results generated in the previous stage. We use issue priority and severity interchangeably within the context of Jira. We apply various hypothesis tests to validate our findings.

In chapter 5, we conclude study and talk about the future of our work and other research opportunities.

Chapter 2

Related Works

2.1 Commit Message Sentiment

Emitza Guzman et al. [12] analyzed sentiment in commit message using SentiStrength tool which uses lexicon based sentiment analysis and tried to figure out relation between sentiment and factors like programming language, day of the week or time of the committed messages, geographical distribution, and project approval. The study revealed several key findings: projects developed in Java tend to have more negative commit comments, whereas those with more geographically distributed teams exhibit higher positive sentiment. Additionally, commit comments written on Mondays are generally more negative. These insights suggest that both the programming language used and the temporal and team-related factors can significantly influence the emotional tone of commit comments in open-source projects.

F.Zibran et al. [16] found a relationship between different express levels(high,low), polarity (positive, negative, neutral), and emotions in bug-introducing and bug-fixing commits using SentiStrength-SE. It is more well trained on software engineering text and gives more accuracy than SentiStrength-SE.The study found that both bug-introducing and bug-fixing commit messages exhibit significantly higher positive emotional scores compared to negative emotional scores. Emotional scores in bug-introducing commits were notably higher during working hours compared to bug-fixing commits. Overall, emotions play a significant role in software development tasks related to bug fixes, with positive emotions prevailing in commit messages.

Ali Zafar Sadiq et al. [15] analyzed four types of commits: Fix-inducing changes

(FICs), parent of FICs (pFIC), Fixing changes (FC) and Fix-inducing fixes (FIFs) and their messages. The Senti4SD classifier is used here.Fix-Inducing Changes (FICs) are significantly more negative than regular commits. Parents of FICs (commits that precede bug-inducing commits) also show more negative sentiment. Correct fixes (commits that successfully fix bugs) are less negative and have more neutral messages compared to incorrect fixes. Bug-related commits have a higher proportion of emotional (non-neutral) messages than regular commits.

2.2 Issue Discussion Sentiment

Kazi Sakib et al. [14] addressed four key research questions focusing on how sentiments in different pull request components relate to FICs. Each component's sentiment (commits, comments, reviews) is individually analyzed to understand its impact on the introduction of bugs.The study discovered a significant correlation between developer sentiment and the likelihood of introducing fix-inducing changes (FICs). It revealed that negative sentiment in developer communications is associated with a higher probability of introducing bugs, whereas positive sentiment tends to correlate with fewer such issues. By analyzing sentiments expressed in pull requests, commit messages, and comments, the research provides robust evidence that the emotional state of developers, as conveyed through their textual communication, directly impacts the quality of code changes.

Marco Ortu et al.[34] investigated the relationship between the sentiment, emotions, and politeness of developers in Jira comments and the time taken to fix issues. Their findings indicate that positive emotions, such as joy and love, are associated with shorter issue fixing times. Conversely, negative emotions, like sadness, correlate with longer issue fixing times. Politeness presents a more complex relationship but generally correlates with longer issue fixing times. Furthermore, the study found that the affective metrics considered (emotion, sentiment, and politeness) show a weak correlation with each other.

Table 3.6 contains a summary of previously addressed research questions from existing literature that are relevant to commit message and issue discussion sentiment.

Paper	**Research Questions**	**Dataset**	**Tool used**

Is Developer Sentiment Related to Software Bugs: An Exploratory Study on GitHub Commits [15]	How does developer sentiment relate to Fix-inducing Changes? How does developer sentiment relate to the parent of Fix-inducing Changes? How does developer sentiment relate to Fixing Changes? How does developer sentiment relate to Fix-inducing Fixes?	GHTorrent	Senti4SD
Understanding the Effect of Developer Sentiment on Fix-Inducing Changes: An Exploratory Study on GitHub Pull Requests [14]	How does sentiment of Commits in Pull Requests relate to Fix-Inducing Changes? How does sentiment of Comments in Pull Requests How does sentiment of Reviews in Pull Requests How does sentiment of all components in Pull Requests relate to Fix-Inducing Changes?	Mined from Github	SentiStrength-SE

Sentiment analysis of commit comments in GitHub: An empirical study [12]	Are emotions in commit comments related to the programming language in which a project is developed? Are emotions in commit comments related to the day of the week or time in which the commits were written? Are emotions in commit comments related to the team geographical distribution? Are emotions in commit comments related to project approval?	Mined from Github	SentiStrength
Sentiment Analysis of Software Bug Related Commit Messages [16]	Do developers express different levels (e.g., high, low) and polarity (i.e., positivity, negativity, and neutrality) of emotions in bug-introducing and bug-fixing commits? Do the developers' polarity (i.e., positivity, negativity, and neutrality) of emotions vary in different times of a day in bug-introducing and bug-fixing commits	Mined from Github	SentiStrength-SE

Exploration and Exploitation of Developers' Sentimental Variations in Software Engineering [17]	Do developers express different levels (e.g., high, low) and polarity (i.e., positivity, negativity, and neutrality) of emotions when they commit different types (e.g., bug-fixing, new feature implementation, refactoring, and dealing with energy and security-related concerns) of development tasks? Can a group of developers be distinguished who express more emotions (positive or negative) in committing a particular type (e.g., bug-fixing) of tasks? Do the developers' polarity (i.e., positivity, negativity, and neutrality) of emotions vary in different days of a week and in different times of a day? Do the developers' emotions have any impact on the lengths of commit comments they write? Do the sizes of the projects and development teams have any impacts on the developers' emotional states at work?	Boa	Modified SentiStrength

Are Bullies more Productive? Empirical Study of Affectiveness vs. Issue Fixing Time [34]	Are emotions, sentiment and politeness correlated to each other? Can developer affectiveness explain the issue fixing Which affective metrics best explain issue fixing time?	Apache Jira	SVM
Analyzing Emotional Contagion in Commit Messages of Open-Source Software Repositories[8]	What is the general sentiment of commit messages amongst different size projects? Do projects with a greater number of commits tend to have a greater number of emotional contagion episodes? Do projects with positive or negative sentimental polarity invoke positive or negative emotional contagion? Does having more members with neutral sentiment polarity result in a greater number of neutral contagion episodes? Do different days of the week have different numbers of contagion effects, i.e. are there more contagious episodes at the start of the week compared to the weekend?	Mined from Github	Wolfram Mathematica software

Table 2.1: Summary of research questions addressed in existing literature relevant to our work.

2.3 Sentiment Analysis Tooling

Prominent rule-based tools in this space include SentiStrength-SE [18], SentiCR [1], and Senti4SD [4]. These are useful for small textual data due to their ability to be tweaked as needed, something which is missing for machine learning models. However, these tools still have the same limitations as other rule-based approaches and are slowly being excluded for newer research [37] [22]. Most machine learning approaches right now utilize some transformer-based architecture due to their ability to self-attention [45]. In particular, architectures based on BERT [7], RoBERTa [23], XLNet [44], and ALBERT [21] consistently score the highest in textual classification tasks such as sentiment analysis [35] [9] [46].

Chapter 3

Proposed Methodology

This chapter contains our methodology and steps to reproduce the results mentioned in the following chapter. We start with the preparation of our working dataset and move on to the metrics used.

3.1 Dataset Preparation

The dataset used for the sentiment analysis in this study is the 20MAD dataset [6]. This dataset provides a large collection of data from two major open-source communities, Apache and Mozilla. It encompasses a vast amount of historical data related to commits and issues over a span of 20 years. The key components of the dataset include:

Table 3.1: Summary of Data

Category	Details
Communities	Apache and Mozilla
Projects	765
Tables	8
Commits	3439001
Issues	2314127
Issue Comments	17303269

The tables of interest to us are the *commits*, *issues*, and *comments* tables. These tables contain textual data that we use for sentiment analysis. In particular, we apply sentiment analysis to the commit messages and issue comments.

The *issues* table contains metadata about issues, such as type and severity. Because our research questions are about issue types and issue severities in the context of Jira, we only use issues that come from Apache who exclusively use Jira for issue tracking. This leaves us with 883065 issues. From these issues we drop all which have null entries in the following columns: *product*, *issue_id*, *priority*, *issuetype*. This leaves us with 870471 issues. From here, we only keep issues with the standard Jira issue priorities: 'Blocker', 'Critical', 'Major', 'Minor', 'Trivial'. This leaves us with 855038 issues.

The *comments* table contains metadata about issue comments. The actual text of the comments is stored in numerous files in a repository-wise fashion. Using regular expression, we find 653 files containing 16861861 sentences from issue discussions from Jira. We merge the sentences back to form 3957479 comments. We join these comments with the 855038 issues previously found and drop any incomplete entries finally giving us 3814162 total issue comments to analyze from 716852 issues.

From the *commits* table, we drop any entries that have null values in the *message*, *committer*, and *repo* columns. This leaves us with 3438700 commits. We then calculated

- **Committer Frequency:** It is discussed in details in section 3.2 . Then we put committers into high , medium and low category.
- **Local Time of Commit:** To find local time of a committer when a commit is made , we used timezone information.
- **Working Times:** We separated working times into four categories morning [6:00–12:00), afternoon[12:00- 18:00), evening [18:00–23:00], and night [23:00-6:00).

We took those products that have greater than 1000 commits.

Some initial preprocessing that we apply to the commit messages and issue comments before sentiment classification is:

- Replacing carriage returns and newline characters with whitespaces
- Removing multiple whitespaces

This gives us our working dataset:

Table 3.2: Summary of working dataset

Table	Count
Commits	3438700
Issue comments	3814162

Finally, we calculated the sentiment class for the commit messages and issue comments.

After finding the sentiment , then we took necessary columns for answering our research questions and made 2 intermediatory tables. All our works have been upload on this [1]

Table 3.4: Table for RQ 3 and RQ 4

commits	
product	*String*
issue_id	*String*
working_times	*String*
committer	*String*
hash	*String*
commit_time	*String*
message_sentiment	*String*
issuetype	*String*
priority	*String*
category	*String*

issue comments	
text_sentiment	*String*
product	*String*
issue_id	*String*
comment_id	*String*
priority	*String*
issuetype	*String*

Overall summary of our final prepared dataset:

Total Products	**Total Committers**	**Total Commits**	**Total Issues**
114	6189	641628	297648

Table 3.5: Summary of Total products,committers,commits, Issues

[1] https://github.com/Siamias202/Exploratory-Analysis-of-Developer-Sentiment-on-Open-Source-Projects

Products	**Commits**	**Committers**	**Issues**	**Bugs**	**Non-Bugs**
HBASE	41661	192	14687	22184	19477
AMBARI	35575	205	21145	26858	8717
SLING	31377	127	6747	9146	22231
CAMEL	26848	206	10458	8956	17892
SPARK	26119	122	15958	11963	14156
SOLR	22598	131	6728	8813	13785
HIVE	18093	160	12709	10767	7326
HDFS	17493	175	8104	7646	9847
LUCENE	17468	110	5615	6573	10895
HADOOP	15805	224	8238	8485	7320

Table 3.6: Top 10 Products based on highest commits

3.1.1 Sentiment Analysis

For sentiment analysis, we use seBERT [29], a BERT$_{\text{LARGE}}$ model pre-trained on a large corpus of textual data from the software domain that won the 2022 Natural Language-based Software Engineering (NLBSE) Tool Competition for text classification tasks [20] [33]. Figure 3.1 visualizes our fine-tuning step.

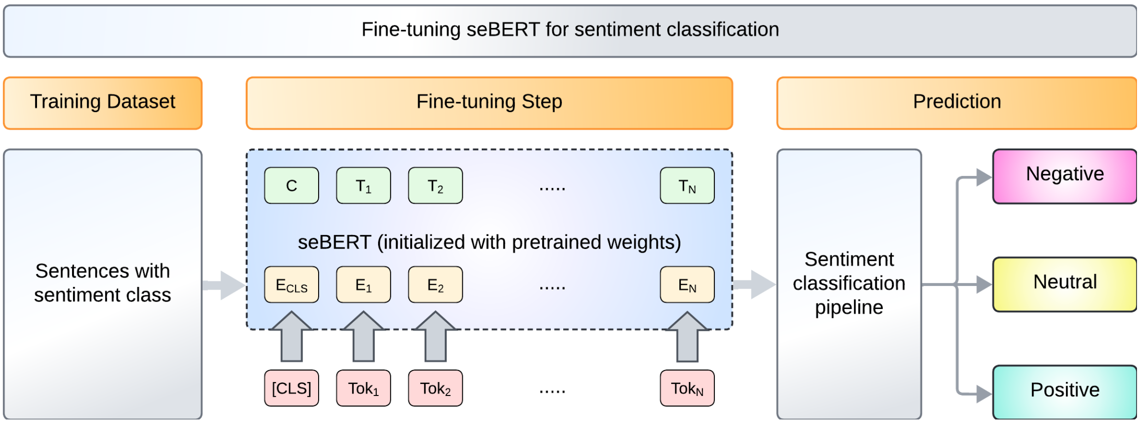

Figure 3.1: seBERT fine-tuning pipeline for sentiment classification task

1. **Training Dataset:** For finetuing, we use a combined dataset that comprises of labeled sentences from different sources. We take away 10% of the data for testing and 20% from the remaining data for validation. The rest is used for training. The dataset is balanced to ensure that the model is not biased towards any particular class.

   - *GitHub Dataset*: This dataset [31] contains sentences from GitHub pull requests and commit comments. This is an important data source as GitHub issues are a common form of communication among developers and so this helps with generalizing the model to classify issue comments from other sources.

   - *Jira Dataset*: This dataset [46] contains sentences from Jira issue comments. This is highly relevant to our task as the trained model is going to be used to classify Jira issue comments later on.

   - *Stack Overflow Dataset 1*: This dataset [3] contains sentences from Stack Overflow questions and answers. This is useful for teaching the model to classify natural conversations between developers. This should help with better classifying comments or commits that fix bugs.

   - *Stack Overflow Dataset 2*: This is another Stack Overflow dataset [46] that contains sentences from questions and answers. A similar labeling guideline was used here compared to the previous datasets.

- *API reviews*: The API reviews dataset [40] contains sentences from API reviews, which are more informal in nature compared to other datasets. This helps the model to classify informal text, or comments from non-developers.

Table 3.7: Sample Distribution for Different Datasets

Dataset	Neutral Samples	Negative Samples	Positive Samples	Total Samples
Github [31]	3022	2087	2013	7122
Jira [46]	0	636	290	926
Stack Overflow 1 [3]	1191	178	131	1500
Stack Overflow 2 [46]	2305	1119	576	4000
API reviews [40]	2635	1048	839	4522
Total samples	**9153**	**5068**	**3849**	**18070**

2. **Fine-tuning:** For training, we apply quidelines from the authors of the original BERT paper [7] and some other studies [38]. The model is first initialized with pretrained weights. We keep the optimizer as AdamW [24] and truncate entries where tokens are larger than 512. We only train for 5 epochs so that the model does not overfit the data as is known to happen for BERT models trained for more epochs. The model with the lowest validation loss is kept for inference [27].

    - Optimizer: AdamW
    - Batch size: 64
    - Epochs: 5
    - Model selection based on lowest validation loss

- Training hardware: RTX 3060 system
- Training time: around 2 hours

Table 3.8: Model evaluation

Metric	Finetuned seBERT	SentiStrength-SE
MCC	79.06%	51.83%
Micro F1	87.16%	71.22%
Macro F1	85.86%	66.52%
F1 (negative)	86.8%	59.35%
F1 (neutral)	89.88%	78.47%
F1 (positive)	80.9%	61.73%
Precision (negative)	88.7%	78.53%
Precision (neutral)	89.31%	68.97%
Precision (positive)	79.95%	73.26%
Recall (negative)	84.97%	47.7%
Recall (neutral)	90.46%	91%
Recall (positive)	81.87%	53.33%

3. **Evaluation:** We evaluate the model using various metrics for negative, neutral, and positive sentiment with our baseline, SentiStrength-SE on the testing split. The results are given in Table 3.8.
4. **Prediction:** We used Hugging Face's sentiment analysis pipeline [13] to calculate the probability scores for positive, negative, and neutral classes. All texts with more than 512 tokens are truncated. The sentiment class with the highest score is kept.

3.2 Regularity Index

To find which developer is more active we developed a regularity index which is used to categorize the developers.
Let,

- D = the total number of days in the project.
- C = the total number of commits by the developer.
- d_i = the number of days the developer made at least one commit.

- c_i = the number of commits made on day i.

Now, Average commits per active day = C/d_i.
This formula helps to understand the typical commit frequency of a developer on the days they are active. It provides a baseline for evaluating daily commit patterns.

To measure the variability in the number of commit made each day we have computed the Standard Deviation of commits per day as below.

$$Commit_{SD} = \sqrt{\frac{1}{d_i}\sum_{i=1}^{d_i}(c_i - \text{average commits per active day})^2}$$

A lower standard deviation indicates a more consistent commit pattern. A higher number of active days d_i relative to the total period D suggests regularity. By combining this two we can define the regularity index R as

$$R = \frac{d_i}{D} \times \frac{1}{(\text{Commit}_{SD} + 1)}$$

$\frac{d_i}{D}$ is the fraction of days the developer was active.

$\frac{1}{(\text{Commit}_{SD}+1)}$ is the inverse of the Standard Deviation of commits per day and we added 1 to avoid the division by zero.

Finally ,

- **Higher R values** indicate developers who are more regular in their commit patterns. This means they consistently make commits on a larger number of days and have less variability in their daily commits.

- **Lower R values** indicate developers who are more irregular in their commit patterns. This means they have fewer active days relative to the total project duration or have a high variability in the number of commits per day.

To find the developers' category :

- We calculated the 30th percentile (P30) and 75th percentile (P75) of the regularity index values.

- Developers with a regularity index below P30 were classified as 'low'.

- Developers with a regularity index above P75 were classified as 'high'.

- The remaining developers were kept in the 'medium' category.

There may be some case where a particular product has less 3 developers. For that

- For 1 developer, they were classified as ’high’.
- For 2 developers, the developer with the higher regularity index was classified as ’high’, and the other as ’low’.

3.3 Emotional Score

In order to compare the issues based on the sentiment values of the discussion comments on the issue, we used some formulae to calculate the emotion scores. For a set I of issue comments belonging to a particular issue where I_+ and I_- are the subsets of comments with positive and negative sentiments, respectively;

Positive Emotional Score, *P(I)* is defined as

$$P(I) = |I_+|$$

Negative Emotional Score, *N(I)* is defined as

$$N(I) = |I_-|$$

Cumulative Emotional Score, *C(I)* is defined as

$$C(I) = P(I) - N(I)$$

Chapter 4

Results and Discussion

In this chapter, we list our findings along with their statistical significance and interpretation.

4.1 RQ1

RQ1: Does Commit frequency affect the developer sentiment on the commit message?

The distribution of sentiments (negative, positive, and neutral) across different committer frequencies (high, low, medium) is summarized in Table 4.1.

Table 4.1: Sentiment distribution across different committer frequencies.

Sentiment	High	Low	Medium
Negative	32721	884	5432
Positive	12184	196	1820
Neutral	495019	12261	81111

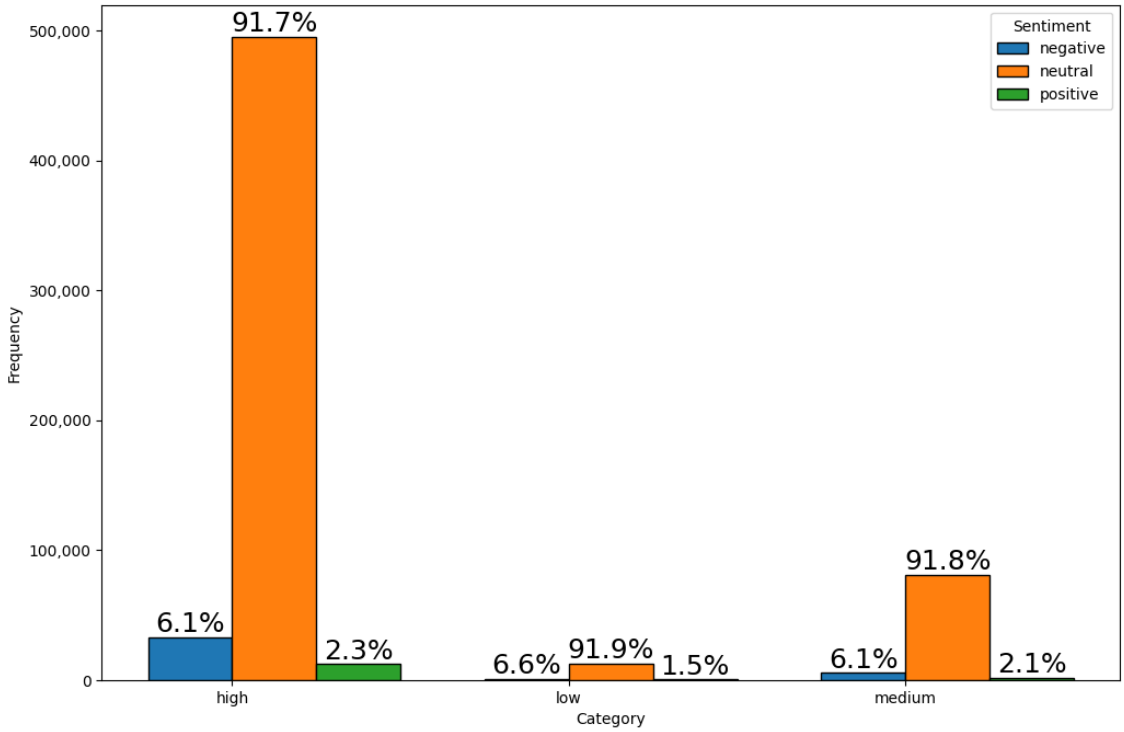

Figure 4.1: Sentiment Frequencies of Commit messages by Committer's Category

4.1.1 Chi-Squared Test Results

A Chi-squared test was performed to determine the association between committer frequency and sentiment counts. The results are as follows:

- **Chi-squared Statistic:** 55.067899563179445
- **p-value:** 3.144×10^{-11}

The expected frequencies for each category are presented in Table 4.2.

Table 4.2: Expected frequencies for each sentiment category across different committer frequencies.

Sentiment	High	Low	Medium
Negative	32849.272769	811.673769	5376.053462
Positive	11949.168054	295.252389	1955.579557
Neutral	495125.559178	12234.073842	81031.366981

Table 4.3: Standardized residuals for Committer Category

Category	Negative	Positive	Neutral
High	-0.707736	2.148267	-0.151437
Medium	0.763030	-3.065889	0.279748
Low	2.538663	-5.776227	0.243438

Here the residual values of low-frequent committer in negative sentiment is higher than others that means it indicates that low committer has more negative sentiment compared to others committers.

4.1.2 Mann-Whitney U test

Table 4.4: Results from conducting Mann-Whitney U test between positive and negative sentiment scores for each committer category

Category	U-statistic	P-value
High	3.053×10^{10}	P«0.05
Medium	7.580×10^{8}	1.261×10^{-163}
Low	1.430×10^{7}	4.900×10^{-31}

Table 4.5: Results from conducting Mann-Whitney U test between positive and negative sentiment scores for each committer category

Category	High	Medium	Low
High	–	0.119	0.081
Medium	0.119	–	0.286
Low	0.081	0.286	–

Answer to RQ1
The analysis reveals a significant association between committer frequency and commit sentiment. Low-category committers have significantly higher negative sentiment than other categories.

4.2 RQ2

RQ2: Does working time affect the developer's sentiment? Is the relation reflected on commit messages?

Distribution of sentiments (negative , positive , neutral) across different working time is summarized in Table 4.6

Table 4.6: Sentiment distribution across different committer frequencies.

Sentiment	Afternoon	Evening	Morning	Night
Negative	17623	8432	10037	2945
Positive	5704	2963	3946	1587
Neutral	243483	133617	147416	63875

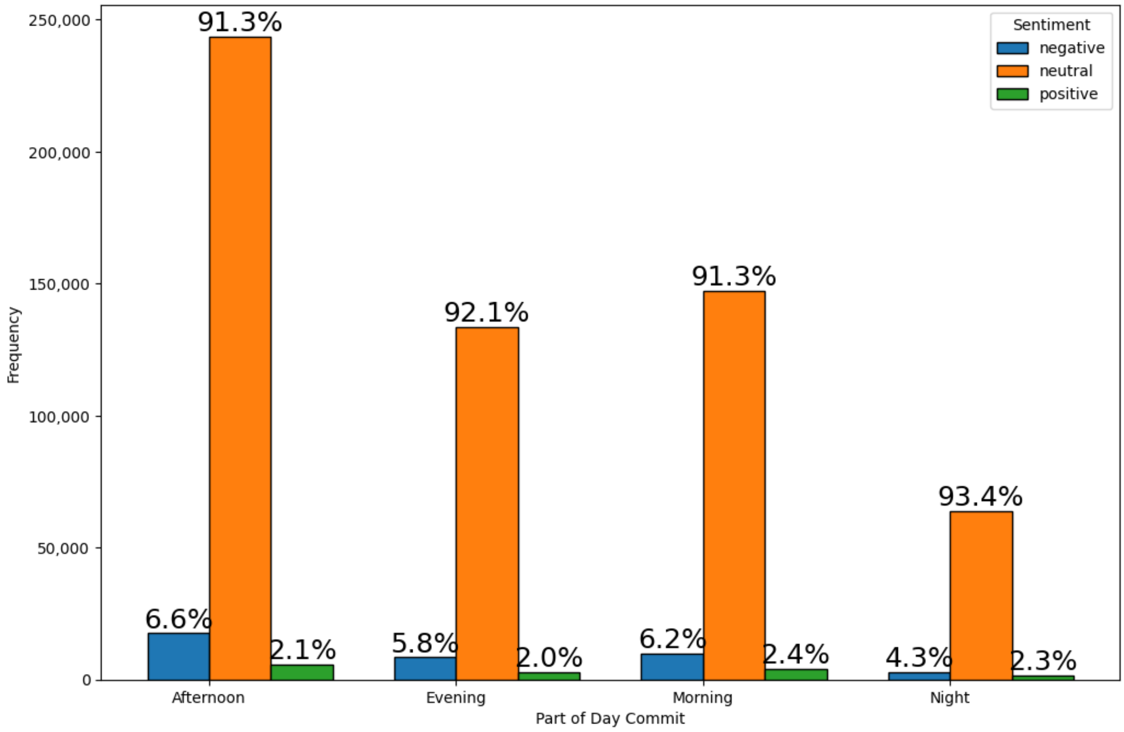

Figure 4.2: Sentiment Frequencies By Working Time

4.2.1 Chi-Squared Test Results

A Chi-squared test was performed to determine the association between working times and sentiment counts. The results are as follows:

- **Chi-squared Statistic:** 597.4521877544643
- **p-value:** 8.266×10^{-126}

The expected frequencies for each working time are presented in Table 4.7.

Table 4.7: Sentiment distribution across different committer frequencies.

Sentiment	Afternoon	Evening	Morning	Night
Negative	16232.866973	8822.609743	9819.603825	4161.919460
Positive	5904.826473	3209.290118	3571.954154	1513.929255
Neutral	244672.306555	132980.100139	148007.442021	62731.151285

Table 4.8: Standardized residuals for Part of Day Commit

Part of Day Commit	Negative	Positive	Neutral
Afternoon	10.910854	-2.613469	-2.404371
Evening	-4.158575	-4.347529	1.746537
Morning	2.193840	6.258524	-1.537342
Night	-18.863184	1.877979	4.566957

Here the residual values of Afternoon in negative sentiment is higher than others that means it indicates that Afternoon has more negative sentiment compared to other parts of the day commit.

4.2.2 Mann-Whitney U test

Table 4.9: Results from conducting Mann-Whitney U test between positive and negative sentiment scores for each part of day

Part of Day Commit	U-statistic	P-value
Afternoon	7.091×10^{9}	P«0.05
Morning	2.667×10^{9}	4.591×10^{-232}
Night	5.301×10^{8}	5.657×10^{-28}
Evening	2.225×10^{9}	4.836×10^{-235}

Table 4.10: Results from conducting Mann-Whitney U test between positive and negative sentiment scores for each part of day

Part of Day Commit	Morning	Afternoon	Evening	Night
Morning	–	4.982×10^{-16}	9.247×10^{-1}	2.587×10^{-25}
Afternoon	4.982×10^{-16}	–	2.136×10^{-15}	2.394×10^{-65}
Evening	9.247×10^{-1}	2.136×10^{-15}	–	7.193×10^{-27}
Night	2.587×10^{-25}	2.394×10^{-65}	7.193×10^{-27}	–

Answer to RQ2
There is minimal association between working times and commit sentiment. Developers working at evening and night tend to show slightly higher negative sentiment.

4.3 RQ3

RQ3: Is the developer's emotion in the issue comment dependent on the issue type, e.g. Bug?

Table 4.11 provides detailed counts of positive, negative, and neutral sentiments for each issue type.

Table 4.11: Sentiment frequencies of issue comments across different issue types

Issue Type	Positive Count	Negative Count	Neutral Count
Bug	467624	376168	1032756
Non-bug	473517	283312	1180785

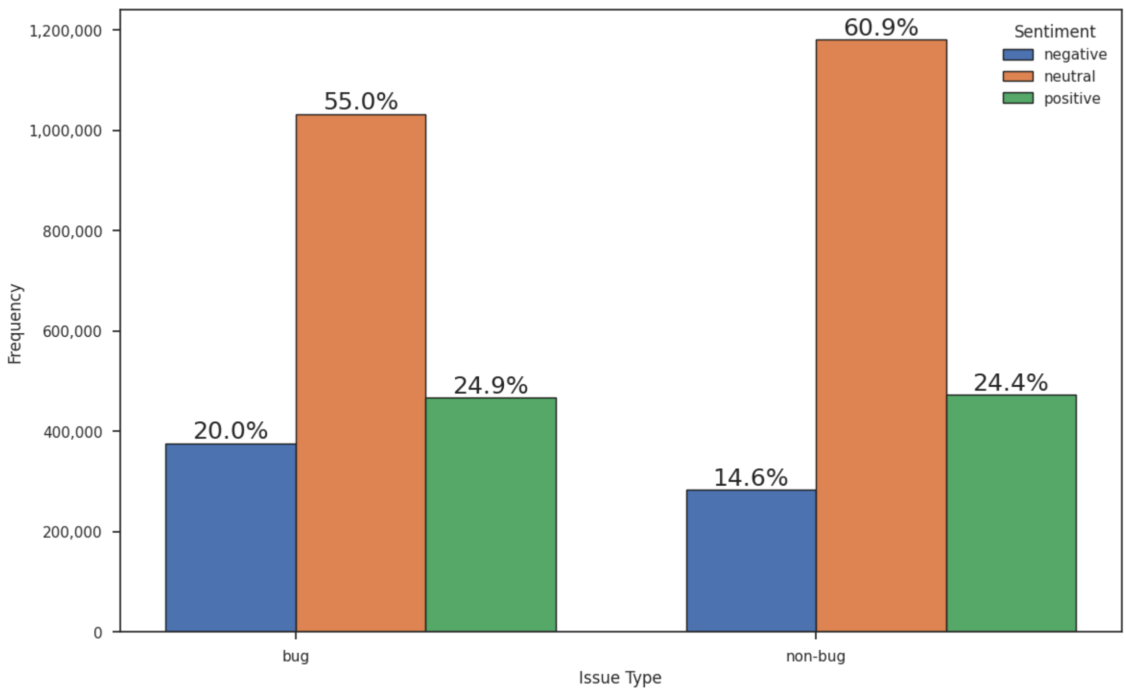

Figure 4.3: Sentiment frequencies of issue comments across different issue types

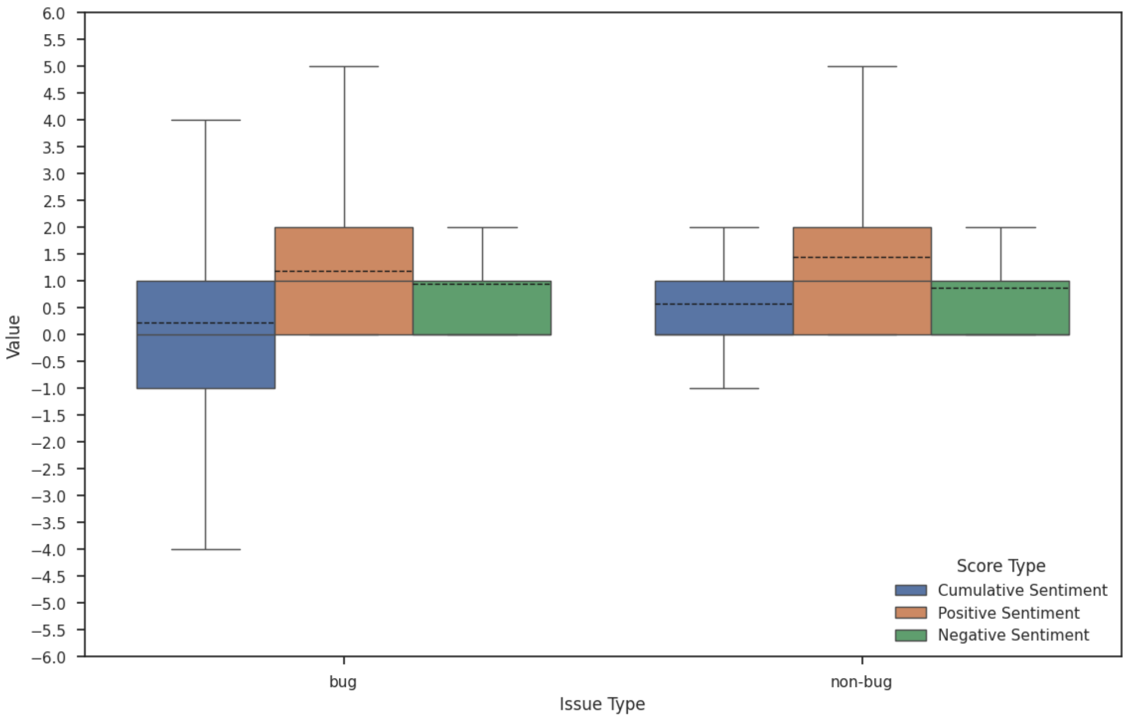

Figure 4.4: Distribution of mean positive, negative, and cumulative sentiment scores of issues for different issue types

4.3.1 Chi Square Test Results

- **Chi-squared Statistic:** 22038.493
- **p-value:** « 0.05

Table 4.12: Expected sentiment frequencies of issue comments across different issue types

Issue Type	Positive Count	Negative Count	Neutral Count
Bug	463036.510	324460.753	1089050.737
Non-bug	478104.490	335019.247	1124490.263

Table 4.13: Standardized residuals for issue types

Issue Type	Positive	Negative	Neutral
Bug	6.742	90.776	-53.944
Non-bug	6.635	-89.334	53.087

4.3.2 Mann-Whitney U test Results

Table 4.14: Results from conducting Mann-Whitney U test between positive and negative sentiment scores for each issue type

Type	**U-statistic**	**P-value**
Bug	8.366×10^{10}	P « 0.05
Non-bug	6.203×10^{10}	P « 0.05

Table 4.15: Results from conducting Mann-Whitney U test (p-values) between cumulative sentiment scores for each pair of issue types

Type	Bug	Non-bug
Bug	—	P « 0.05
Non-bug	P « 0.05	—

Answer to RQ3

Issue type significantly influences issue discussion sentiment. Issues related to bugs are associated with more negative sentiment, and so they show a wider range of cumulative emotional scores compared to issues that are not related to bugs.

4.4 RQ4

RQ4: Do the emotions in the issue comments differ depending on the severity of the issue?

Table 4.16 provides detailed counts of positive, negative, and neutral sentiments for each issue severity.

Table 4.16: Sentiment frequencies of issue comments across different severities

Severity	**Blocker**	**Critical**	**Major**	**Minor**	**Trivial**
Negative	39716	49265	439509	117301	13689
Neutral	102122	134894	1531149	393634	51742
Positive	44046	55221	638294	177164	26416

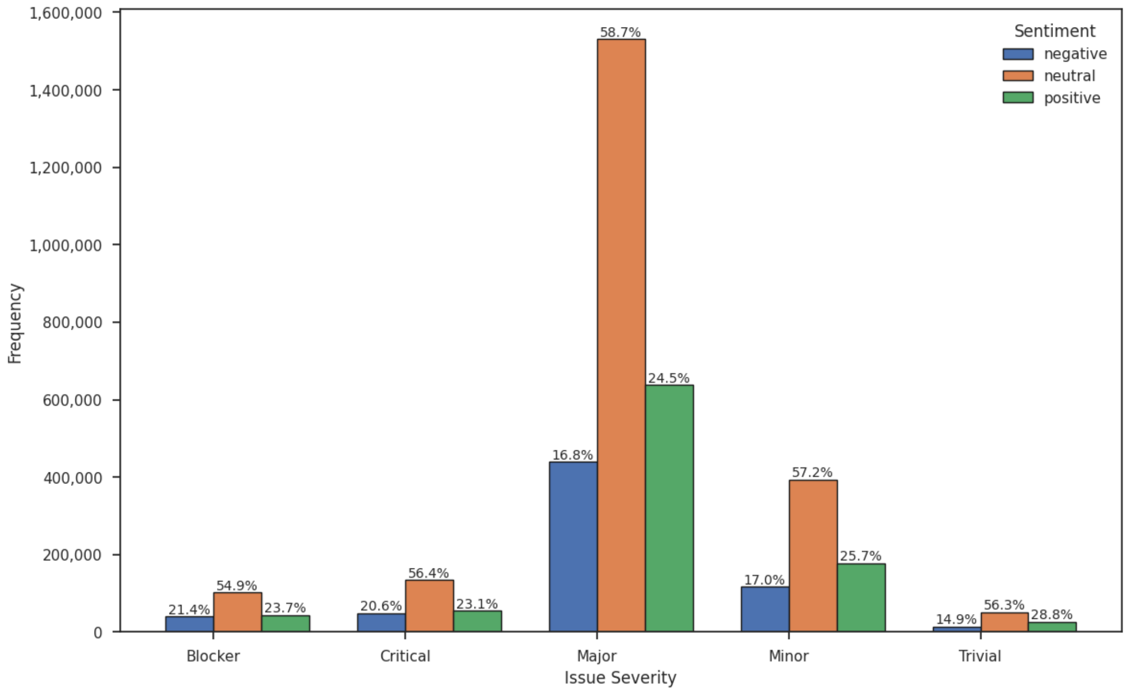

Figure 4.5: Sentiment frequencies of issue comments across different severities

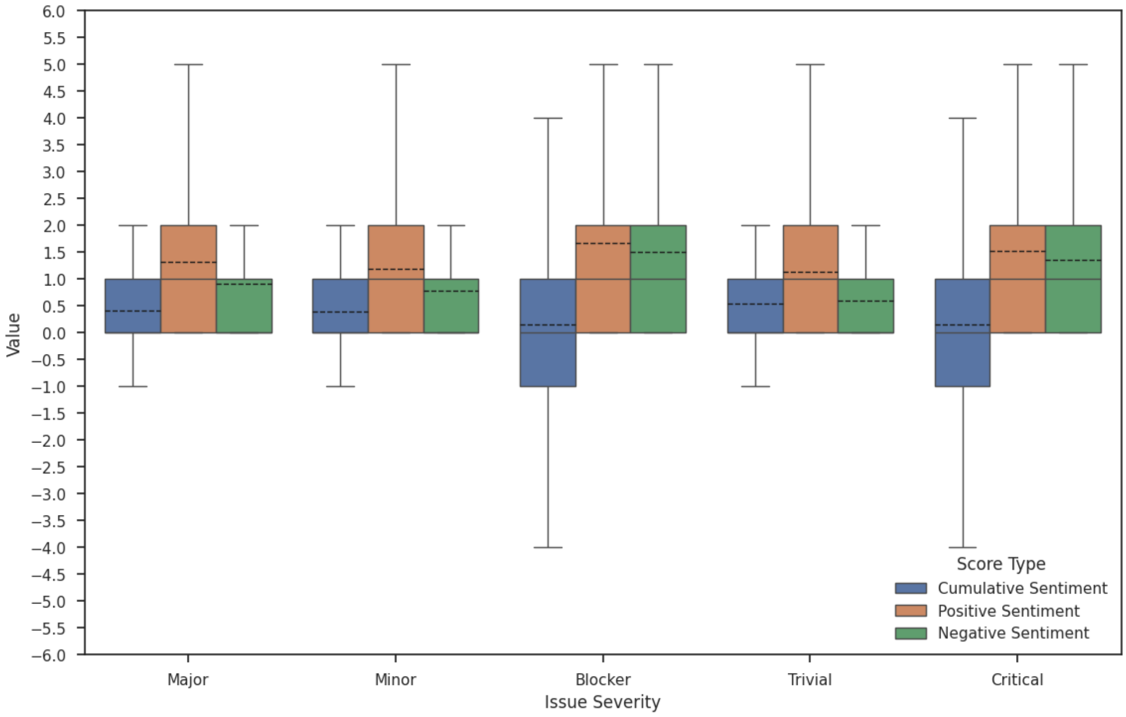

Figure 4.6: Distribution of mean positive, negative, and cumulative sentiment scores of issues for different severities

4.4.1 Chi Square Test Results

- **Chi-squared Statistic:** 5961.983

- **p-value:** « 0.05

Table 4.17: Expected sentiment frequencies of issue comments across different severities

Severity	**Blocker**	**Critical**	**Major**	**Minor**	**Trivial**
Negative	32139.899	41389.517	451095.592	118974.372	15880.621
Neutral	107877.394	138923.686	1514099.878	399336.826	53303.216
Positive	45866.708	59066.797	643756.530	169787.802	22663.164

Table 4.18: Standardized residuals for issue severities

Severity	**Blocker**	**Critical**	**Major**	**Minor**	**Trivial**
Negative	42.259	38.711	-17.251	-4.851	-17.391
Neutral	-17.523	-10.811	13.856	-9.024	-6.762
Positive	-8.501	-15.824	-6.808	17.901	24.929

4.4.2 Mann-Whitney U test Results

Table 4.19: Results from conducting Mann-Whitney U test between positive and negative sentiment scores for each issue severity

Severity	**U-statistic**	**P-value**
Blocker	3.593×10^{8}	5.765×10^{-15}
Critical	6.856×10^{8}	1.245×10^{-28}
Major	1.313×10^{11}	P « 0.05
Minor	1.271×10^{10}	P « 0.05
Trivial	3.379×10^{8}	P « 0.05

Table 4.20: Results from conducting Mann-Whitney U test (p-values) between cumulative sentiment scores for each pair of issue severities

Severity	Blocker	Critical	Major	Minor	Trivial
Blocker	—	5.036×10^{-1}	4.083×10^{-92}	5.000×10^{-93}	5.102×10^{-193}
Critical	5.036×10^{-1}	—	1.413×10^{-115}	2.917×10^{-113}	2.570×10^{-223}
Major	4.083×10^{-92}	1.413×10^{-115}	—	2.329×10^{-2}	8.469×10^{-113}
Minor	5.000×10^{-93}	2.917×10^{-113}	2.329×10^{-2}	—	9.787×10^{-97}
Trivial	5.102×10^{-193}	2.570×10^{-223}	8.469×10^{-113}	9.787×10^{-97}	—

Answer to RQ4
Issue severity also significantly influences issue discussion sentiment. Blocker and critical severity issues are associated with more negative sentiment, while trivial issues tend to be associated with high positive sentiment.

4.5 Threats to Validity

4.5.1 Internal Validity

The internal validity of this work depends on the sentiment output of seBERT. Our fine-tuned seBERT outperforms the rule-based SentiStrength-SE on our training dataset, although it cannot give 100% accurate results in determining the emotional polarity of a given commit message. Additionally, due to the lack of datasets for labeled commit messages, our training dataset does not contain labeled commit messages, so the output of the emotional polarity of commit messages can be improved greatly if such a dataset is supplied.

4.5.2 Construct Validity

One question that may arise concerning the validity of the categorization of the period of day in which developers made commits because the developers may be physically located at different geographic locations and time zones. So, we used the timezone information associated with the commit times to find the local commit time of the corresponding developers.

4.5.3 External Validity

The findings of this work based on our study on more than 3.4 million commit messages and 3.8 million issue comments. This amount is a huge amount of data that enhances the generability of our results. However, they do not represent the current status of the software development landscape due to the dataset being a few years old.

4.5.4 Reliablity

The methodology of data collection, data preprocessing and sentiment output of commits messages are well documented in this paper. All the codes are available in our github repository. Hence, it should be possible to replicate the study.

Chapter 5

Conclusion

This study explored several dimensions of developer sentiment as reflected in commit messages and issue comments, focusing on the impact of various factors such as committer frequency, working time, issue type, and issue severity. Through rigorous statistical analyses, we derived several key insights:

- **Committer Frequency:** There is a significant association between committer frequency and sentiment. High committer frequency is associated with a higher proportion of negative sentiments compared to low and medium frequencies.
- **Working Time:** The results indicate no statistically significant difference in negative sentiment counts between commits made during different parts of the day although ones made during evening and night have very slightly more cases of negative sentiment.
- **Issue Type:** The study found a significant dependency of developer emotion in issue comments on the issue type. Issues identified as bugs elicit a much higher negative sentiment in comments compared to other types of issues.
- **Issue Severity:** There is a statistically significant association between the severity of issues and the sentiment expressed in issue comments. Blocker and critical severity issues are associated with more negative sentiment, while trivial issues tend to be associated with high positive sentiment.

These findings contribute to a deeper understanding of how various factors influence developer sentiment, which can have implications for project management, team dynamics, and overall software quality. By identifying the conditions that lead to higher negative sentiments, project managers and team leaders can implement strategies to mitigate negative emotions, improve developer morale, and enhance productivity.

Future research could further explore the causality behind these associations and examine additional factors that may influence developer sentiment. Moreover, longitudinal studies could provide insights into how sentiment evolves over time and in response to different interventions.

Overall, this study highlights the importance of considering emotional factors in software development and offers valuable insights for improving the well-being and effectiveness of development teams.

References

[1] T. Ahmed, A. Bosu, A. Iqbal, and S. Rahimi, “Senticr: A customized sentiment analysis tool for code review interactions,” in *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE ’17, Urbana-Champaign, IL, USA: IEEE Press, 2017, pp. 106–111, ISBN: 9781538626849.

[2] I. E. Asri, N. Kerzazi, G. Uddin, F. Khomh, and M. Janati Idrissi, “An empirical study of sentiments in code reviews,” *Information and Software Technology*, vol. 114, pp. 37–54, 2019, ISSN: 0950-5849. DOI: `https://doi.org/10.1016/j.infsof.2019.06.005`.

[3] E. Biswas, M. Karabulut, L. Pollock, and K. Vijay-Shanker, “Achieving reliable sentiment analysis in the software engineering domain using bert,” in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2020, pp. 162–173. DOI: `10.1109/ICSME46990.2020.00025`.

[4] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, “Sentiment polarity detection for software development,” *Empirical Softw. Engg.*, vol. 23, no. 3, pp. 1352–1382, Jun. 2018, ISSN: 1382-3256. DOI: `10.1007/s10664-017-9546-9`.

[5] D. Che, M. Safran, and Z. Peng, “From big data to big data mining: Challenges, issues, and opportunities,” in *Database Systems for Advanced Applications: 18th International Conference, DASFAA 2013, International Workshops: BDMA, SNSM, SeCoP, Wuhan, China, April 22-25, 2013. Proceedings 18*, Springer, 2013, pp. 1–15.

[6] M. Claes and M. V. Mantyla, “20-mad - 20 years of issues and commits of mozilla and apache development,” in *2020 IEEE/ACM 17th International Conference on Mining Software Repositories (MSR)*, Los Alamitos, CA, USA: IEEE Computer Society, May 2020, pp. 503–507. DOI: `10.1145/3379597.3387487`.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[8] R. Dhakad and L. Benedicenti, “Analyzing emotional contagion in commit messages of open-source software repositories,” *Natural Language Processing and Machine Learning*, 2023.

[9] A. Di Sorbo and S. Panichella, “Summary of the 1st natural language-based software engineering workshop (nlbse 2022),” *ACM SIGSOFT Software Engineering Notes*, vol. 48, no. 1, pp. 101–104, 2023.

[10] P. Ekman, “An argument for basic emotions,” *Cognition & emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[11] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, “What happens when software developers are (un)happy,” *Journal of Systems and Software*, vol. 140, pp. 32–47, 2018, ISSN: 0164-1212. DOI: `https://doi.org/10.1016/j.jss.2018.02.041`.

[12] E. Guzman, D. Azócar, and Y. Li, “Sentiment analysis of commit comments in github: An empirical study,” *11th Working Conference on Mining Software Repositories, MSR 2014 - Proceedings*, May 2014. DOI: `10.1145/2597073.2597118`.

[13] *Hugging face pipelines*, `https://huggingface.co/docs/transformers/en/main_classes/pipelines`.

[14] S. Huq, A. Sadiq, and K. Sakib, “Understanding the effect of developer sentiment on fix-inducing changes: An exploratory study on github pull requests,” in *2019 26th Asia-Pacific Software Engineering Conference (APSEC)*, Los Alamitos, CA, USA: IEEE Computer Society, Dec. 2019, pp. 514–521. DOI: `10.1109/APSEC48747.2019.00075`.

[15] S. F. Huq, A. Sadiq, and K. Sakib, “Is developer sentiment related to software bugs: An exploratory study on github commits,” Feb. 2020, pp. 527–531. DOI: `10.1109/SANER48275.2020.9054801`.

[16] M. R. Islam and M. F. Zibran, “Sentiment analysis of software bug related commit messages,” *Network*, vol. 740, p. 740, 2018.

[17] M. R. Islam and M. F. Zibran, “Exploration and exploitation of developers’ sentimental variations in software engineering,” vol. 4, no. 4, pp. 35–55, Oct. 2016, ISSN: 2166-7160. DOI: `10.4018/IJSI.2016100103`.

[18] M. R. Islam and M. F. Zibran, “Sentistrength-se: Exploiting domain specificity for improved sentiment analysis in software engineering text,” *Journal of Systems and Software*, vol. 145, pp. 125–146, 2018, ISSN: 0164-1212. DOI: `https://doi.org/10.1016/j.jss.2018.08.030`.

[19] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian, "The promises and perils of mining github," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014, Hyderabad, India: Association for Computing Machinery, 2014, pp. 92–101, ISBN: 9781450328630. DOI: 10.1145/2597073.2597074.

[20] R. Kallis, O. Chaparro, A. D. Sorbo, and S. Panichella, "Nlbse'22 tool competition," in *2022 IEEE/ACM 1st International Workshop on Natural Language-Based Software Engineering (NLBSE)*, Los Alamitos, CA, USA: IEEE Computer Society, May 2022, pp. 25–28. DOI: 10.1145/3528588.3528664.

[21] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[22] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto, "Sentiment analysis for software engineering: How far can we go?" In *Proceedings of the 40th International Conference on Software Engineering*, ser. ICSE '18, <conf-loc>, <city>Gothenburg</city>, <country>Sweden</country>, </conf-loc>: Association for Computing Machinery, 2018, pp. 94–104, ISBN: 9781450356381. DOI: 10.1145/3180155.3180195.

[23] Y. Liu, M. Ott, N. Goyal, *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[24] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[25] L. Marks, Y. Zou, and A. E. Hassan, "Studying the fix-time for bugs in large open source projects," in *Proceedings of the 7th International Conference on Predictive Models in Software Engineering*, ser. Promise '11, Banff, Alberta, Canada: Association for Computing Machinery, 2011, ISBN: 9781450307093. DOI: 10.1145/2020390.2020401.

[26] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014, ISSN: 2090-4479. DOI: https://doi.org/10.1016/j.asej.2014.04.011.

[27] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černockỳ, "Strategies for training large scale neural network language models," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, IEEE, 2011, pp. 196–201.

[28] A. Mitra and S. Mohanty, "Sentiment analysis using machine learning approaches," *Journal of Ubiquitous Computing and Communication Technologies (UCCT)*, vol. 2, no. 03, pp. 145–152, 2020.

[29] J. von der Mosel, A. Trautsch, and S. Herbold, "On the validity of pre-trained transformers for natural language processing in the software engineering domain," *IEEE Transactions on Software Engineering*, vol. 49, no. 04, pp. 1487–1507, Apr. 2023, ISSN: 1939-3520. DOI: 10.1109/TSE.2022.3178469.

[30] A. Murgia, P. Tourani, B. Adams, and M. Ortu, "Do developers feel emotions? an exploratory analysis of emotions in software artifacts," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014, Hyderabad, India: Association for Computing Machinery, 2014, pp. 262–271, ISBN: 9781450328630. DOI: 10.1145/2597073.2597086.

[31] N. Novielli, F. Calefato, D. Dongiovanni, D. Girardi, and F. Lanubile, "Can we use se-specific sentiment analysis tools in a cross-platform setting?" In *2020 IEEE/ACM 17th International Conference on Mining Software Repositories (MSR)*, Los Alamitos, CA, USA: IEEE Computer Society, May 2020, pp. 158–168. DOI: 10.1145/3379597.3387446.

[32] N. Novielli, F. Calefato, and F. Lanubile, "The challenges of sentiment detection in the social programmer ecosystem," in *Proceedings of the 7th International Workshop on Social Software Engineering*, ser. SSE 2015, Bergamo, Italy: Association for Computing Machinery, 2015, pp. 33–40, ISBN: 9781450338189. DOI: 10.1145/2804381.2804387.

[33] M. Obaidi, L. Nagel, A. Specht, and J. Klünder, "Sentiment analysis tools in software engineering: A systematic mapping study," *Inf. Softw. Technol.*, vol. 151, no. C, Nov. 2022, ISSN: 0950-5849. DOI: 10.1016/j.infsof.2022.107018.

[34] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli, "Are bullies more productive? empirical study of affectiveness vs. issue fixing time," in *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, 2015, pp. 303–313. DOI: 10.1109/MSR.2015.35.

[35] S. Panichella and A. Di Sorbo, "Summary of the 2nd natural language-based software engineering workshop (nlbse 2023)," *ACM SIGSOFT Software Engineering Notes*, vol. 48, no. 4, pp. 60–63, 2023.

[36] J. A. Russell, "A circumplex model of affect.," *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[37] M. U. Sarwar, S. Zafar, M. W. Mkaouer, G. S. Walia, and M. Z. Malik, "Multilabel classification of commit messages using transfer learning," in *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2020, pp. 37–42. DOI: 10.1109/ISSREW51248.2020.00034.

[38] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" In *Chinese computational linguistics: 18th China national conference,*

CCL 2019, Kunming, China, October 18–20, 2019, proceedings 18, Springer, 2019, pp. 194–206.

[39] A. Trautsch and S. Herbold, "Predicting issue types with sebert," in *Proceedings of the 1st International Workshop on Natural Language-Based Software Engineering*, ser. NLBSE '22, Pittsburgh, Pennsylvania: Association for Computing Machinery, 2023, pp. 37–39, ISBN: 9781450393430. DOI: `10.1145/3528588.3528661`.

[40] G. Uddin and F. Khomh, "Automatic mining of opinions expressed about apis in stack overflow," *IEEE Transactions on Software Engineering*, vol. 47, no. 03, pp. 522–559, Mar. 2021, ISSN: 1939-3520. DOI: `10.1109/TSE.2019.2900245`.

[41] A. Valdez, H. Oktaba, H. Gómez, and A. Vizcaíno, "Sentiment analysis in jira software repositories," in *2020 8th International Conference in Software Engineering Research and Innovation (CONISOFT)*, 2020, pp. 254–259. DOI: `10.1109/CONISOFT50191.2020.00043`.

[42] C. Wang, Y. Li, L. Chen, W. Huang, Y. Zhou, and B. Xu, "Examining the effects of developer familiarity on bug fixing," *Journal of Systems and Software*, vol. 169, p. 110 667, 2020, ISSN: 0164-1212. DOI: `https://doi.org/10.1016/j.jss.2020.110667`.

[43] A. Yadav, S. K. Singh, and J. S. Suri, "Ranking of software developers based on expertise score for bug triaging," *Inf. Softw. Technol.*, vol. 112, no. C, pp. 1–17, Aug. 2019, ISSN: 0950-5849. DOI: `10.1016/j.infsof.2019.03.014`.

[44] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[45] T. Zhang, I. C. Irsan, F. Thung, and D. Lo, "Revisiting sentiment analysis for software engineering in the era of large language models," *arXiv preprint arXiv:2310.11113*, 2023.

[46] T. Zhang, B. Xu, F. Thung, S. A. Haryono, D. Lo, and L. Jiang, "Sentiment analysis for software engineering: How far can pre-trained transformer models go?" In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Sep. 2020, pp. 70–80. DOI: `10.1109/ICSME46990.2020.00017`.