Effects of Personality Traits and Emotional Factors in Pull Request Acceptance.

by

Rahul N. Iyer

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2019

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Social interactions in the form of discussion are an indispensable part of collaborative software development. The discussions are essential for developers to share their views and to form a strong relationship with other teammates. These discussions invoke both positive and negative emotions such as joy, love, aggression, and disgust. Additionally, developers also exhibit hidden behaviors that dictate their personality. Some developers can be supportive and open to new ideas, whereas others can be conservative. Past research has shown that the personality of the developers has a significant role in determining the success of the task they collaboratively perform.

Additionally, previous research has also shown that in online collaborative environments, the developers use signals from comments such as rudeness to determine if they are compatible to work together. Most of these studies use traditional small-scale surveys for their experiments. The transparent nature of online collaborative environments makes it easier to conduct empirical experiments by mining pull request comments. In this thesis, first, we investigate the effect of different personality traits on pull request acceptance. The results of this experiment will provide us with a valuable understanding of the personality traits of developers and help us develop tools to assist developers. We follow it with a second experiment to understand the influence of different emotional factors on pull request decisions. The emotion expressed by a developer on their teammates can be influenced by social statuses, such as the number of followers. Moreover, the teammate's team status, such as team member or outside contributor too, can influence the emotional effect. To understand moderation, we investigate different interaction effects.

We start the experiment by replicating Tsay et al.'s work that examined the influence of social factors (e.g., 'social distance') and technical factors (e.g., test file inclusion) for evaluating contributions. We extend their work by augmenting it with personality traits of developers and examining the influence of on the pull request evaluation process in GitHub. In particular, we extract the 'Big Five' personality traits (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism) of developers from their online digital footprints, such as pull request comments. We analyze the personality traits of 16,935 active developers from 1,860 projects and compare their relative importance to other non-personality factors from past research, in the pull request evaluation process. We find that pull requests from authors (requesters) who are more open and conscientious, but less extroverted, have a higher chance of approval. Furthermore, pull requests that are closed by developers (closers) who are more conscientious, extroverted, and neurotic, have a higher likelihood of acceptance. The larger the difference in personality traits between the requester and the closer, the more positive effect it has on pull request acceptance.

Although the effect of personality traits is significant and comparable to technical factors, we find that social factors are still more influential when it comes to the effect in the likelihood of pull request acceptance.

We perform a second experiment to analyze the effect of emotions on pull request decisions. To predict emotions in the comments, we develop a generalised, software engineering specific language model that outperforms previous machine learning algorithms on four different standard datasets. We find that the percentage of positive comments from both requester and closer has a positive association with pull request acceptance, whereas the percentage of negative comments has a negative association. Also, the polarity of the emotion associated with the first comment of both requester and closer had a positive association with pull request acceptance, i.e., more positive the emotion, the higher the likelihood of acceptance. Finally, we find that social factors moderate the effects of emotions.

Acknowledgements

I want to begin by thanking my supervisors Prof Jesse Hoey and Prof Meiyappan Nagappan without their support this research would not have been possible. Their constant guidance and feedback have been essential for my learning and progress. I would also like to thank Prof Michael Godfrey and Prof Olga Vechtomova for agreeing to read my thesis and providing valuable feedback.

I want to especially thank my collaborators Josh, Alex Y, and Prof Daniel Vogel. I would also like to thank all the members of THEMIS-COG project.

Finally, I want to thank my family and friends I made here in Waterloo for their unwavering support and encouragement. Special thanks to my friends Josh, Alex Y, Alex Sachs, Angshuman, Lakshmanan, Dishant, Nalin, Hemant, and Vineet for useful discussions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

GitHub is an online collaborative environment that uses pull-request-based development where developers work collectively to improve projects. In this process, the developers (requesters) 'fork' the project (i.e., make a personal copy of the project) and make changes to their personal copies of the code to add functionality. Next, the developers can request project managers (closers) to merge the changes they have made in their personal copies to the main branch in the form of a pull request. The project manager then evaluates this pull request on different parameters and decides to either merge (accept) or close (reject) the pull request.

The closer and the requester interact during the evaluation process to reach a consensus over the acceptance of the contribution. They often leave comments on the code commits and pull request threads to clarify any misunderstandings or suggest further changes. Apart from the requester, other developers join the conversation, sharing their views on the contribution. Each participating developer leaves a digital footprint visible to other collaborators. This footprint includes one's code quality, coding style, activeness in the project, and the comments they leave on GitHub. GitHub provides the ability to view anyone's profile and see their follower count, which represents social respect, and their personal repositories that portray technical skills and open source activeness.

Previous studies on open source systems have consistently shown that the closer's decision to accept a contribution not only depends on the technical quality of the contribution, but also on underlying social impressions [26]. Developers use social signals, such as 'social distance' (e.g., a requester who follows the closer has a social distance of one), while evaluating the pull request [87]. The social signals used by closers were found to be equally as important as technical factors such as test file inclusion, lines of code modified, and

files changed. In addition to social signals, some impressions are directly visible to other developers in the form of comments, displaying different sentiments and behaviours that form the user's underlying personality [65, 37]. Another study suggests that closers look at person-based factors like previous interactions with the developer when they are uncertain about the value of the contribution [54]. Impolite or argumentative comments suggest negative 'personality' and such developers are deemed difficult to work with. Recently, a top Linux developer's behaviour was found to have negative effects on the open source community, which may have pushed a lot of volunteer contributors away from the open source development [2].

Given that non-technical factors, such as interpersonal skills, are important, it is reasonable to believe that understanding developers' personality can provide valuable information about the group dynamics and the success of the project. Personality, by definition, seeks to make predictions about what individuals do in given situations [21]. In reverse, we can observe how developers have behaved in a given project, extract their personality traits, and examine its influence on the pull request acceptance. Given the diverse community of GitHub, we expect to see many developers exhibiting different behaviours, characterized by varying personality traits. Some developers may have strict and high standards of coding, while others may have relaxed standards for acceptance. Similarly, some may be encouraging and readily available to assist outside contributors, while others reject pull requests immediately. With the rich data available on GitHub, understanding the behaviours of developers has become easier.

A study by Ng et al. [67] suggests that emotions expressed by a person regulate the person's personality. Past research in software engineering has indicated that emotions play an essential role in productivity, requirement engineering, collaboration, and feedback cycle [65, 37, 45]. Emotions in the comments, unlike personality, are easy to interpret and consequently can provide valuable information about the state of the pull request to the developer. A negative reply from the project manager [1](closer) can be interpreted as unhappiness or disagreement concerning the pull request. Similarly, project managers can also gain confidence over the pull request by understanding the emotions of requester's comments.

Thesis Statement: *Personality and emotional factors have a significant association with pull request decisions.* Each developer in an online collaborative environment has a different personality. We believe a developer with a personality has a better likelihood of pull request acceptance, and a project manager with a specific personality accepts more

[1]Note: The term 'Project managers' and 'closers' are used interchangeably throughout the thesis. Similarly, 'requesters' and 'contributors' are used interchangeably.

pull requests than others. Additionally, we believe emotion associated with a pull request is predictive of the pull request acceptance. By understanding the effects of personality and emotion, we can develop tools to help developers better interpret the current state of the pull request and suggest better ways to respond.

In this thesis, to validate the statement we conduct different empirical experiments. We start of by extending the work of Tsay et al. [87] to examine the role of developers' personality traits. The personality traits are derived from the Five-Factor Model (FFM) or the 'Big Five' [24], which consists of five traits: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. The main question we address is: "Does the personality of a developer affect pull request acceptance?". This may be addressed by the following research questions:

- **(RQ0) Can Tsay et al.'s results be replicated on a more recent dataset?**
- **(RQ1) Does the personality of a requester influence the likelihood of the pull request being accepted?**
- **(RQ2) Does the personality of a closer influence the likelihood of the pull request being accepted?**
- **(RQ3) Does a difference in personality between requester and the influence the likelihood of the pull request being accepted?**

To understand the effect of emotions on pull request acceptance, we performed another experiment. We also conduct further experiments to evaluate the interaction effects between social factors and emotions. We address the following research questions:

- **(RQ1) Does the percentage of positive and negative emotions influence pull request acceptance?**
- **(RQ2) Does the percentage of positive and negative emotions of the requester influence pull request acceptance and moderation effect caused by other social factors?**
- **(RQ3) Does the emotion of closers first response aid in predicting pull request acceptance and moderation effect caused by other social factors?**
- **(RQ4) Does the percentage of positive and negative emotions of the closer influence pull request acceptance and moderation effect caused by other social factors?**

- **(RQ5) Does the emotion of closers first response aid in predicting pull request acceptance and moderation effect caused by other social factors?**

1.1 Contributions

In summary, the primary contributions of this paper are:

1. Replication of Tsay et al.'s [87] work to show generalisability.
2. Empirical evidence showing the effects of requesters' and closers' personality traits, and the difference in personality traits between the requester and the closer on pull request acceptance decision making.
3. Supporting and explaining the evidence from literature in personality psychology.
4. Empirical analysis of the emotional effects of requester and closer's comment on pull request acceptance and different interaction effects that moderate emotion's predictive power.
5. We develop a state of the art sentiment classification model that outperforms previous methods on 4 standard datasets.

Chapter 2

Background and Related Works

2.1 GitHub and pull request evaluation process

Github is a code hosting platform which makes collaborative software development easier. The site offers free public and private repository hosting. GitHub is a default choice for most of the organization that want to host open source projects due to the high number of active developers in the platform. This makes it easy for organizations to invite external developers to collaborate. Collaboration is done by submitting a pull request to add new features or solve bugs in the code. The medium can also be used to raise issues in the code or feature requests with the help of active issue tracker. According to Github Octoverse [5], Github is home to more than 31 million developers, more than 2 million organizations, more than 96 million repositories and has over 200 million pull requests. GitHub as a collaborative coding environment makes a lot of information apart from the code transparent. This gives the developers ability to follow other users, similar to following a user on Twitter, to get and see updates from the them. The developers can also star a repository that is equivalent to bookmarking a repository. The users can also watch a repository in which new activities in the repository are notified to the user. Developers also have their own profile page where they can list their personal information such as their email address, where they work and city of residence. Additionally, profile page also has information on their repositories and contribution activity. Developers can use the information to get a better understanding of their teammates. They can look at how active a developer is, understand the code quality and also get a sense of their behavior through their comments. These non technical transparent features influence pull request acceptance [87]. An experiment by Tsay et al. found social factors such as prior interaction of the

requester in the project and social connection were predictive of pull request acceptance.

Github uses pull requests as the standard way of contributing code to a project. To start contributing to a project, a developer needs to fork the repository i.e create a personal copy of the repository and make changes on it. Once the developer completes making changes to the code, they can use the git version control system to commit changes to his personal copy. As a final step the developer submits a request to the original repository to accept the changes in the form of a pull request. The project manager can then choose to accept or reject the contribution. Pull requests can be in one of the 3 stages: open, merged/accepted or closed/rejected. An open pull request suggests the contribution is still under consideration and no decision has been made yet. In this stage developers discuss the quality of code and express their thoughts on the pull request through comments. A closed/rejected pull request is the stage when the project manager decides to reject the contribution. The cause of rejection can be due to disagreement between contributor and maintainer or failing test cases or even bad coding style. Finally, a merged/accepted indicates an accepted contribution. The code changes submitted by the contributor has now been merged with the repository.

Github provides a REST API service to extract many useful information regarding project owner, repository and the user. A REpresentational State Transfer (REST) is a service that uses HTTP protocol to provide information to the user. The protocol consists of four operations - GET to retrieve, PUT to update, POST to insert and DELETE to remove an element. Github allows GET operation on all the public repository and allows the other three only if the user is authorised to access the repository. For some of the entities, the API service only provides the final state information, meaning any changes that happened between the intial and final is not stored. For example, when we request the follower count of a user the system only returns the current followers. There is no way to query the system to get the follower count of a user at a specific time. To overcome this difficulty, GHTorrent system was designed by Gousios [34]. GHTorrent is a database that stores the meta-data of the information exposed by the GitHub event stream.

2.2 Pull Request Evaluation

The first empirical study in evaluating the influence of different factors on pull request acceptance was conducted by Tsay et al. [87]. They showed that project managers not only use technical factors, but also use social clues while evaluating pull requests. Prior interactions inside the project and 'social distance' were important to the pull request acceptance process. Gousios et al. [35] conducted a large-scale analysis of factors affecting

pull request acceptance on 1.9 million pull requests. The results reaffirmed the existence of non-technical factors involved in pull request evaluation process. Soares et al. [83] showed that the speed of the pull request evaluation process is significant in pull request acceptance. Yu et al. [92] studied different factors that explained the latency of a pull request evaluation. They recognized continuous integration and first human response time as important factors.

2.3 Personality Theory

The study of personality has a rich history with many perspectives, but at its core, it is concerned with human nature and patterns of behaviour [40]. It seeks to explain why there exist individual differences in behaviour and to predict how one would behave in a given situation. Personality psychologists have been interested in developing a unified theory of personality and are concerned with the validity of its measurements [32].

Theorists naturally gravitated towards categorizing people into 'types'. In particular, the Jungian personality types received a lot of attention and are thought to be measured by the Myers–Briggs Type Indicator (MBTI) questionnaire [66]. Despite its popularity, the validity of MBTI has met with criticisms [59, 75].

One of the most influential personality models is the Five-Factor Model (FFM) or the 'Big Five' [24]. It is comprised of several traits or subpersonalities and they collectively explain one's disposition and behaviour. The Big Five is empirically well validated, showing good reliability, validity [58] and consistency across cultures [32]. We adopt the definitions of traits from the 10 Aspects scale [31]:

- *Openness to Experience:* A measure of intellect and openness. Individuals who are high on Openness to Experience enjoy solving complex problems and show aesthetic appreciation. Hereafter, Openness to Experience will be referred to as Openness.
- *Conscientiousness:* A measure of industriousness and orderliness. Individuals who are high on Conscientiousness are detail-oriented and reliable workers.
- *Extraversion:* A measure of enthusiasm and assertiveness. Individuals who are high on Extraversion are gregarious and take charge in social situations.
- *Agreeableness:* A measure of compassion and politeness. Individuals who are high on Agreeableness are nurturing and trusting of others, and uncomfortable confronting others.

- *Neuroticism:* A measure of volatility and withdrawal. Individuals who are high on Neuroticism are prone to experiencing negative emotions have a lot of self doubts.

2.4 Study of Personality in Software Engineering

Past research has focused on evaluating the developers' personality traits using the MBTI. Karn and Cowling [48] examined the association between the personality profile of teams and their performance. The results showed teams that were more heterogeneous in personality worked better together than homogeneous teams. They also reported some evidence of personality clashes among the members that produced intense debates, resulting in no progress. Capretz and Ahmed [20] mapped software engineering tasks to specific MBTI traits using skills required for the tasks and determined which traits would be useful for a team. Although there has been plenty of research on modelling personalities in the software domain using MBTI, Varona et al. [88] found many inconsistencies among different works.

Researchers have started to utilize the Big Five measures to study the effects of personality in the software domain. Acuna et al. [7] examined the relationships of personality traits with team processes, task characteristics, product quality, and team satisfaction. They found that teams with high job satisfaction tend to have members who score highly on Agreeableness and Conscientiousness. Martinez et al. [56] conducted an empirical case study which suggested relationships between certain Big Five personality traits and software engineering roles.

With advancements in language processing and availability of different NLP tools, researchers have started deriving personality traits from raw text using different psycholinguistic tools instead of interviews and surveys. Rigby and Hassan [81] studied the personality of developers in the Apache httpd server mailing list using the Linguistic Inquiry and Word Count (LIWC) dictionary. Their results suggested that two top developers responsible for the major Apache releases had similar personalities that were different from the other developers. Bazelli et al. [14] conducted analyses on the personality traits of StackOverflow users using the LIWC dictionary. They retrieved the personality traits from the raw text which included questions asked and answers given by the users. Their results revealed that top reputed authors in StackOverflow tend to be more extroverted than other users. Licorish et al. [50] profiled personalities of developers across the globe for the IBM jazz repository. The results showed that top contributors tend to score high on Openness, practitioners involved in usability tasks tend to score high on Extraversion, and coders

tend to score high on Neuroticism. Rastogi and Nagappan [80] also used the LIWC dictionary to find the personalities of Github developers. Their results showed that the top contributor was dramatically different in that they were significantly more neurotic than other contributors.

Newly developed methods are taking advantage of machine learning and are outperforming rule-based methods that use the LIWC dictionary [9]. A recent work by Paruma et al. [73] used *IBM Watson Personality Insights* (we used the same method) to retrieve personality traits for clustering developers together. They found a relationship between the personality traits of committers and their social and technical activities on a project. Calefato et al. [16] analyzed the developers' personality in Apache projects using *IBM Watson Personality Insights*. They observed that developers became more conscientious, agreeable, and neurotic over time and found no significant evidence of contributor's team membership affecting their personalities. They also noted that developers who are more open and more agreeable have a better chance of becoming project contributors. Our work bares some resemblance to Calefato et al.'s work [16] in that we also examine the effects of personality traits on the likelihood of becoming project contributors. However, it is different, as we try to perform the experiment on the data from GitHub pull requests and at a much larger scale. We also position the personality factors with other social and technical factors that help us better gauge the relative importance of personality traits in pull request evaluation process. Additionally, we look at the closer's personality traits and also the difference in personality traits between the requester and the closer.

We extract personality scores of the developers with the help of IBM Personality Insights. We provide developer's comments on the GitHub platform which includes issue comments, pull request comment and commit comments to the IBM tool and it provides a percentile score between 0 and 1 for each personality trait indicative of the relative standing in the community. More details on the experiment is provided in the later sections.

2.5 Study of Sentiment and Emotion in Software Engineering.

Research in the past decade has focused on assessing the sentiments and emotions of developers from different mediums like Stack Overflow, GitHub, App Reviews, and Twitter. Brooks et al. conducted one of the first work on analysing sentiments and emotions of online communication logs of developers. The authors performed analysis on 485,000 chat messages of developers in Factory Chat Dataset. The graduate students annotated the

chat comments with 13 different affect lexicon. The authors then use machine learning algorithms like SVM, Naive Bayes, and tree-based algorithms to scale up the classification. Guzman and Bruegge [38] used LDA topic modeling to extract topic summaries from the text communication of a software development team and assigned sentiments to the topics with the help of SentiStrength. They evaluated their approach from 1000 collaborative artifacts derived from emails and web pages of 3 different projects. Interviews with project leaders suggested correlations between emotions and the state of the project. Another study by Guzman et al. [37] involved an empirical analysis on commit comments to understand the relationship of sentiments with factors such as time of day, day of the week, distribution of the team and commit approval. The authors analysed 60425 commits comments from 90 different projects and assigned sentiment scores to each comment with the help of SentiStrength. They observed Java projects to have comparatively more negative scores than C, C++, JavaScript, PHP, Python, and Ruby. They also find 78 percent of comments were written during the weekdays while just 20 percent had been written on the weekends. Additionally, the authors also conclude comments that were written on Monday's produced more negative comments. Finally, their analysis showed that developers from specific continents and countries were more positive while commenting on the Github platform.

Murgia et al. [65] performed an exploratory study to analyze if software engineering artifacts like issue comments have emotional information and do humans use emotional information. The goal of the authors was to develop a fully automatic emotion classification tool. The authors use Tier 1 parrot's framework: love, sadness, anger, joy, surprise, and fear for emotion categories. To experiment, the authors retrieved all the issues and their comments from the JIRA issue tracking system of Apache Software Foundation and sampled a small subset from around 271,000 comments for annotation. A group of 4 master and Ph.D. students were paired together in 2 different groups for annotation. The authors concluded that software artifacts contain emotional content with specific emotions like love and surprise targeting co-workers, whereas fear targets artifacts, respectively. Authors also found more disagreement among the human raters on specific emotions where highest disagreement was found when there was no emotion in the artifact. The context of the comment did not have a significant affect on the agreement, where, on the contrary, more disagreements were noticed.

Pletea et al. [76] analysed the sentiments associated with security related discussions on GitHub. The authors examined 60,658 commit comments and 54,892 pull request discussions and then determined if the comment was security related using a set of security keyword. The authors used Python's Natural langauge ToolKits native sentiment classifier to classify each comment and found that security related discussion had more negative emotion. Another emiprical study was proposed by Novielli et al. [68] that explored the

role of emotional style of questions asked on Stack Overflow and its relationship with the probability of answers the post gets. They analysed more than 7 million posts and termed a question to be successful who had at least one accepted answer. Affective classes in the post were assigned with the help of Linguistic Inquiry Word Count dictionary. Dewan [30] proposes a symbiotic relationship between emotions and software engineering. A relationship where emotion detection makes collaboration more meaningful, focused and to have implicit information related to the feedback from team members. On other hand, the author also says software engineering can help make reusable tools which do automated analysis based on the emotions.

Sinha et al. [82] analyzed the sentiments of commit logs of 28000 projects over the span of 7 years. The authors analysed more than 2.1 million non-empty commit messages and used SentiStrength tool to assign each commit message to a sentiment value. They found 18 percent of all the commit messages to be negative while only 7 percent where found to be positive and majority rest being neutral. Similar to the Guzman et al. [37], the authors find that the sentiment of commits written on Monday had more positive and more negative sentiments. Tuesday was found to have the most negative commits on average and Friday with most positive commits. Lastly, the authors found positive correlations between the number of files changed and the sentiment value associated with the commit messages.

Destfanis et al. [29] analyzed politeness in comments of 22 software projects developed using agile board of JIRA repository. To measure the politeness score, the authors used the politeness tool provided by Danescu et al. [27] which computes scores as a binary value : +1 for polite and -1 for impolite. They then inferred the overall politeness of a particular issue by grouping its associated issue comments together. Grouping them as polite, if all comments were polite, impolite if all the comments were impolite and mixed if there were a mix of polite and impolite comments. The authors found time to fix an issue is shorter the the issue is polite compared to impolite and mixed issues. Additionally, the authors also find attractiveness of a project and ability to keep hold of developers was correlated with the percentage of polite comments. Finally, the authors also noted the politeness in comments varies over time and bugs with lower priority had less politeness compared to the bugs with higher priority.

Ortu et al. [71] annotated 2000 issue comments and 4000 sentences written by developers with affective categories such as love, joy, surprise, anger, sadness, and fear. They aim to provide a platform for affective research in the software engineering community. Calefato et al. [17] presented a preliminary empirical design framework to measure if affective trust derived from social communication between developers has a positive association with successful collaborations. They also used SentiStrength to assign sentiment scores to

comments. A study by Marshall et al. [55] aimed at understanding the effect of emotions in student teams in a software engineering course found that posts with fewer emotions performed better and less affective individuals were evaluated more positively.

Another study by Islam et al. [44] anlaysed 490k comments from 50 open source projects to find emotional variations in different types of activities a developer performs. They used SentiStrength to classify the comments into sentiments and augmented with a manually curated list of software engineering specific terms which can be misleading. The authors find developers express positive and negative emotions almost equally in tasks which require more work whereas they find positive emotions expressed more in tasks which involved bug fixes. The authors were also able to distinguish developers who were mostly positive, negative, and others who used both positive and negative emotions. The authors could not get significant results for emotional variations at different times of day and days in the week. Lastly, authors were seen to be posting longer commit messages when they are emotionally more active. An empirical study by Destafanis et al. [28] involved analysing 370,000 comments from around 100,000 issues and 25,000 users in Github for affective emotions. The authors extracted sentiments, emotions, and politeness in the comments using different pre-built tools. The result showed that commenters (non-contributor) were less polite and more harmful than the users. They also find the commenters used fewer emotions than users but had higher levels of valence, arousal, and dominance.

In this thesis, we measure the effects of emotions to predict the pull request decisions significantly different from the works discussed above. We start by exploring the emotion of the pull request thread by modeling the percentage of positive and negative emotions. Secondly, we model the percentage of positive and negative emotions of the requester and closer separately. Also, we try to understand the information contained in the first response of both requester and closer. Finally, we also explore how other social factors moderate the emotional effects by introducing interaction effects.

2.6 Tools for analysing emotions in Software Engineering text.

SentiStrength: SentiStrength[86] is a tool for calculating the sentiment associated with texts. It is a rule-based algorithm that uses a sentiment dictionary of words and different linguistic patterns to calculate the sentiment of the sentence. The tool assigns a positive and a negative score for each word. Each word receives 2 scores, one between -1 and -5 representing negative sentiment and other1 to 5 representing positive sentiment. Sen-

tistrength tool functions on the principle that a sentence can express both negative and positive emotions. Thus, the tool provides both positive and negative scores for each word. The tool calculates the sentiment of the sentence as the sum of the maximum score of positive sentiment and a minimum score of negative sentiment. Apart from the absolute scores, the tool also can report binary (-1,+1) and trinary (-1,0,+1) categories.

SentiStrength-SE: SentiStrength[86] is generic tool for extracting sentiment values from natural language. SentiStrength does not take into account the domain-specific words, which can add many emotions to the overall sentence. SentiStrength-SE [45] is a modified version of SentiStrength with different set of dictionary. It is built on top of SentiStrength and uses the same underlying API. SentiStrength-SE was seen to outperform SentiStrength when evaluated by the authors.

Senti4SD: Senti4SD [18] is a sentiment polarity classifier trained on 4,000 StackOverflow questions, answers and comments. Senti4SD uses custom word embedding trained on StackOverflow data as base features to exploit semantic and lexicon features. The word embedding was trained using CBOW architecture for different embeddings sizes of 400,600,800 and 1000 with a vocabulary size of 360,000. The authors use the Support Vector Machine algorithm to train the dataset. Senti4SD also provides the annotated dataset along with a tool to help with the development of the more models. The dataset has well balanced 3 categories negative, neutral, and positive. The tool was found to improve the classification accuracy compared to SentiStrength and SentiStrength-SE.

SentiCR: SentiCR [8] is another tool to assign sentiments trained specifically on code review comments. Unlike Senti4SD, this tool uses a simple bag of words features derived from TF-IDF. The TF-IDF features are simple word statistic derived by multiplying the word frequency by the log of inverse document frequency ratio of the word. This scheme assigns higher scores to rare words and smaller scores to words that appear frequently. Before generating the feature vector, the tool uses prepossessing steps like stemming, stop word removal, and raw code removal. The tool also uses the SMOTE up-sampling technique to generate fake examples to solve the data imbalance problem. The tool uses a Gradient Boosting Trees algorithm to train the model. The authors report that the model outperforms many rule-based and simple machine learning-based models.

Emotxt: EmoTxt [19] is a tool developed to classify software engineering text into emotions based on WordNet emotions category (love, joy, anger, sadness, fear, and surprise). The tool also provides a gold standard dataset of 4800 StackOverflow comments, questions, and answers. The tool was trained using one vs. all binary Support Vector Machine models.

SentiSW: SentiSW is a tool created by Ding et al. [33] to classify issue comments from

Github. They trained their dataset on 3000 issue comments and found 68 percent mean precision, which outperforms other tools. The tool also returns entities along with the sentiment, which helps in identifying the entities that invoke a particular sentiment.

Other Tools: Many researchers have used different tools apart from the tools mentioned above. Researchers have used the Linguistic Inquiry Word Count dictionary, Bing Liu Lexicon of Opinion Words dictionary, NRC Emotion Lexicon, and WordNet Affect dictionary. People have also tried using the inbuilt NLTK sentiment classifier. Ram et al. [79] tried using modern deep learning architectures to assign sentiments and had limited success in a few datasets.

Although there are enough tools available to classify the text into sentiment, the software engineering community has identified significant inconsistencies in the way the emotions are expressed in the text. These inconsistencies pose a significant challenge in developing unified models across the domain. Novielli et al. [69] outlines many problems with sentiment analysis tools and pushes for a more granular look at the affective states than just the polarity. Jongeling et al. [46] conducted an empirical analysis of the performances of different sentiment classification tools on 7 different datasets. The authors observed significant disagreement within the tools which lead to improper conclusions. The authors found that a tool trained on specific dataset could not generalize on other datasets, which implies that the linguistic use is different for each task. Hence we need a more general representation as the model can not learn diverse linguistic variation with small samples of data. On the contrary, another study by Novielli et al. [70] concluded that even though there are disagreements between the tools, its minimal and is dataset dependent. A study by Lin et al. [51] concluded that sentiment analysis tools are unreliable and are not able to discriminate neutral text from positive and negative texts reliably.

The above reasons prompted us to think in the direction of unified domain-specific language models, that capture the context, and with fine-tuning can reliably predict the sentiment categories. The next section explains the need for language models and provides the necessary background for our experiment.

2.7 Software Engineering Specific Language Models

While, we have 4 sentiment classification datasets in software engineering, as discussed above, there is not enough labeled data to train a model properly. For the model for understanding the language correctly, we require a large sample of labeled data. The understanding of language is primal to classifying sentiments, classifying issues category,

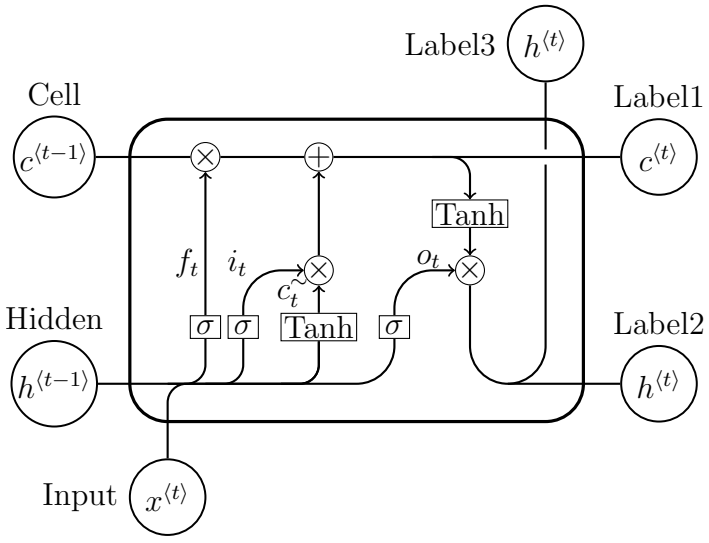


Figure 2.1: LSTM Architecture

and bug severity labels. On the contrary, there exists a vast amount of unlabeled text data in the form of posts and comments on StackOverflow or other mediums such as Github. Recent advances in natural language processing have suggested training a language model using the unlabeled dataset and then doing task-specific fine-tuning on the labeled one improves the performance [78]. Pre-training involves training a language model to predict the next word given a phrase. During the training process, the model learns many nuances of language using large and diverse unlabelled dataset. Task-specific fine-tuning involves replacing the last layer of the language model, that predicted the next word, with a task-specific classification layer and tuning the weights slowly. The section first describes the basics of the neural network, language model, and model architecture used in this thesis. Later, in the following chapter, we describe our experiment on StackOverflow and the model's performance on different standardised datasets. Finally, we end the section by providing the performance of the model on Github comments.

2.7.1 Basics

LSTM

Recurrent neural Networks (RNNs) learn representations of sequences over time but they have been found to not have ability to retain important memory over the time. RNNs tend

to forget the information from the past very quickly, and as these information are essential in the downstream task they tend to perform poorly over longer sequences. To mitigate the problem of forgetting, Long Short Term Memory cells (LSTM) were introduced by Hochreiter and Schmidhuber [39]. LSTMs improve the memory by explicitly modelling what to forget and what to remember in the memory cell. The cell structure is comprised of 3 main gates :– input gate, output gate and forget gate. All the gates consist of a sigmoid activation function which squashes the activation between 0 and 1 which helps in regulating the information flow. LSTM also has a cell state which acts as the memory and interacts with gates to retain or forget the information.

The mathematical equation of the LSTM is

$$
\begin{aligned}
c_t^{\sim} &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
c_t &= f_t \odot c_{t-1} + i_t \odot c_t^{\sim} \\
h_t &= \tanh(c_t) \odot o_t
\end{aligned}
$$

f_t is the output of the forget gate, i_t is the output of the input gate, c^sim_t is the output after applying tanh to the hidden and input values and o_t is the output of the output gate. c_t is the cell state which is responsible for holding memory and h_t is the current hidden representation.

Dropout

Dropout is a regularization technique proposed by Srivatsava et al. [84]. The main idea is to randomly drop the output of neurons in a layer i.e. set the output of the neurons to zero in the forward pass. This acts as a regularization as it prevents the network from over fitting to the dataset by relying on only fewer neurons to generate higher level representations. The number of neurons that are dropped is controlled with a probability parameter 'p' where higher value means higher probability of a neuron being dropped. Using dropout has empirically been shown to improve the performance of all neural network models.

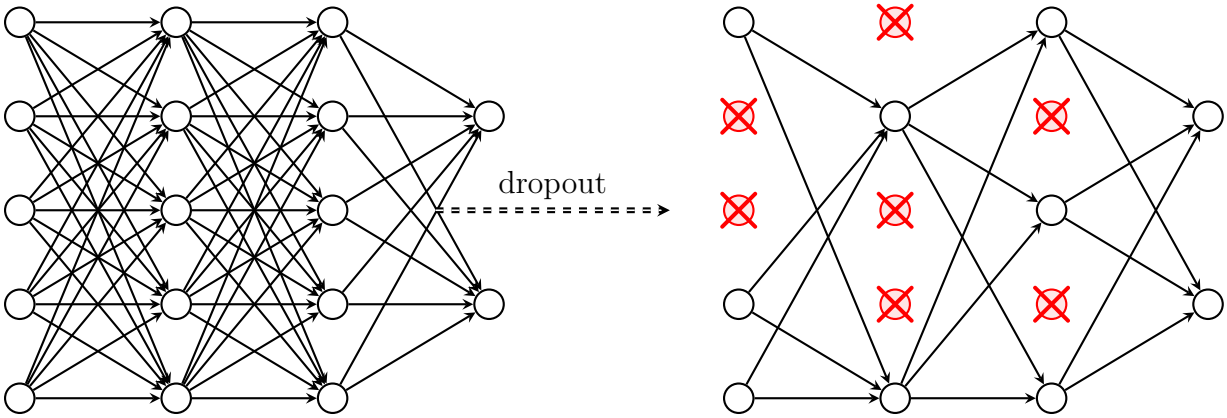


Figure 2.2: The figure shows DropOut. The output of crossed neurons are set to zero.

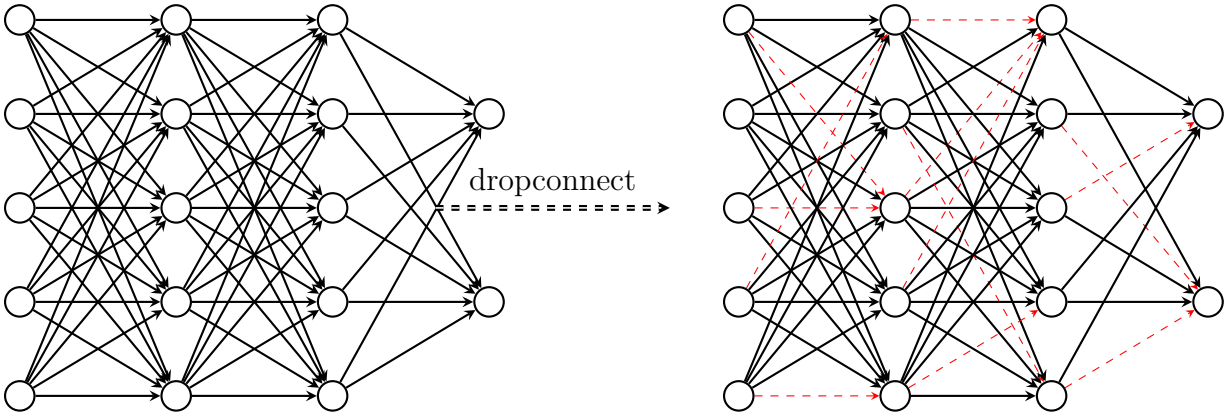


Figure 2.3: The figure shows DropConnect. The red dashed lines show the dropped connections or weights.

DropConnect

DropConnect was proposed by Wan et al [89] as generalisation of Dropout. Unlike Dropout, DropConnect drops the connection of a neuron randomly, i.e it makes weight between 2 neurons to 0 but since a neuron has other connection still alive the output is non zero. Each connection between 2 layers of neural network can now be dropped with a probability of p. DropConnect is especially used in recurrent neural networks where the network needs to store information about the past for prediction. In Dropout we make the activation as 0, hence reducing the ability of the network to hold previous information but in DropConnect we dilute this information and not make it completely zero.

Language Models

Language models are form of probabilistic models that predict next word or character. Given a sequence of words, language model predicts the next word in the sequence. The language model outputs a probability distribution over the vocabulary. To get a good model, language models are trained on large and diverse set of texts such Wikipedia. Language model forms a fundamental base for many natural language processing tasks because to perform a specific task, an understanding of the language is important. A simple statistical language model's equation can be written as

$$P(w_1, ..., w_m) = \prod_{i=1}^{m} P(w_i | w_1, ..., w_{i-1})$$

This equation can be approximated by only considering the last 'n' words which makes computation a lot easier and hence this model is called n-gram language model.

$$P(w_1, ..., w_m) = \prod_{i=1}^{m} P(w_i | w_{i-n}, ..., w_{i-1})$$

Recently, Neural language models have started to gain popularity over the conventional statistical language models. Neural language models was first proposed by Bengio et al [15] in 2003. The model is usually implemented using a variant of recurrent neural network like LSTM. These models combined with word embeddings outperform the conventional n-gram models.

2.7.2 AWD - LSTM

The AWD LSTM architecture consists of a simple multi-layer LSTM network with many regularization features that improve the performance of an LSTM significantly. The regularization features involves dropouts applied in different parts of the network. Different types of Dropout used in the network is explained as follows:

Embedding Dropout

Embedding Dropout involves applying dropout on the embedding matrix. The dropout mask is applied at the word level and zeros are broadcast to all the dimensions of the specific word embedding. The non dropped words are scaled by $1/(1-p_e)$ where p_e is the embedding dropout probability. The scaling is performed to keep the sum of the embedding constant during the training and inference time.

Input Dropout

This dropout is applied on the specific word embedding before it is passed as an input to the LSTM. This dropout uses a special variant of dropout called variational dropout. In a traditional dropout system, a new dropout mask is generated on every time sequence. On contrary, variational dropout uses the same mask throughout the entire sequence. This forces the model to learn representation without relying on the masked dimensions of the hidden vector.

Hidden Weight Dropout

Merity et al. [63] proposed a novel way to enforce Dropout to regularize the model with the help of DropConnect. From the LSTM equation, W_{**} are the Weight matrices from different gates., t is the time step and $\odot$ is the dot product. This dropout uses DropConnect on all hidden to hidden weights (W_{h*}. This helps prevent over-fitting on the network as the weights remain dropped for all the time step in forward as well as backward phase as the weights are shared across different time sequence. This acts as regularization, as the model needs to rely on specific set of weights for all time sequences.

Hidden Output Dropout

Hidden output dropout uses variational dropout on the output of the LSTM that feeds as an input to the higher layers of LSTM. This dropout is not applied to the last layer of LSTM and is only used between different LSTM layers.

Encoder Output Dropout

Output dropout also uses variational dropout scheme on the output of the last LSTM Layer. This is applied just before the outputs of the final hidden representation is passed to the decoder for next word prediction.

Universal Language Model Fine-tuning (UlmFit)

Ulmfit system was proposed by Howard and Ruder [43]. The system involves training a general domain training of a language model and then using fine-tuning to adapt the language model for other classification tasks to prevent over-fitting even in datasets with extremely small examples. The purpose of pre-traning language model is to capture long term dependencies, hierarchical relations and other hidden features like sentiment. This model works universally across all tasks, uses same architecture and requires no feature engineering. The process involves training a training a general language model, fine-tuning the language model for the task and then using transfer learning for the classification task. The model outperforms in many classification datasets like ImDB, Trec, DbPedia and Yelp. The system consists of:

- AWD-LSTM based language model
- Target Task LM finetuning
- Fine-tuning with gradual unfreezing

The experiment details and training process is described in the further chapters.

Chapter 3

Analysing the effects of personality traits of developers in pull request acceptance.

Many researchers in the software engineering community have focused on understanding the personality of the developers in the workplace. There has been significant growth in the number of open sources projects in the past decade, and additionally, there has not been enough research conducted to understand the personality and behavior of developers in online collaborative environments. In this chapter, we examine the effects of personality traits of the pull request decision. We start by describing the methodology used in the experiment. Later, we motivate and formulate different research questions. Finally, we summarise the results and discuss the findings.

3.1 Methodology

In this section, we describe our data selection, data extraction, and data modelling process.

3.1.1 Data Selection

We considered valid projects from two different sources. The first set of projects is the 11,000 projects used by Tsay et al. [87]. The second source was taken from a set of more than one million projects curated by Muniah et al. [64]. The dataset contains project meta

data including a binary label on project maintenance. The dataset is publicly available on their project website *RepoReaper* [6]. We filtered the *RepoReaper* dataset down to 15,000 projects and kept projects that:

- Were properly maintained which excludes defunct projects, such as student course projects and/or casual projects uploaded by developers.
- Had more than ten issues. This removes projects that have not received a good amount of attention. Issues are usually raised when one notices bugs or requests a feature to be implemented.
- Had at least three contributors. We want projects to have received some contributions from volunteer developers.
- Were not forks. Duplicate copies of the project were not considered.

From a pool of 26,000 projects, we filtered it to 1,860 projects by only considering those projects that have at least 250 closed or merged pull requests to ensure good amount of discussions in the project for the *IBM Watson Personality Insights* to extract the personality traits. We then extracted the pull request data of 1,860 projects using GHTorrent [34], GitHub API [3], and a modified version of the open source code provided by Gousios et al. [35]. The data and the model building script can be found here [1].

3.1.2 Obtaining Personality Traits

We used the *IBM Watson Personality Insights* [4] to retrieve personality of the developers from raw text. In addition to the Big Five personality traits, this service can be used to extract other information such as developers' needs and underlying values. The service uses Arnoux et al.'s [9] open vocabulary approach that applies Gaussian Processes on GloVe Word Embeddings to infer the personality [74]. The authors were able to achieve similar or better accuracy with 8 times less data. This method outperforms the previous rule-based methods that used LIWC dictionary [4]. Recent works have utilized the *IBM Watson Personality Insights* to compute personality traits [47, 73, 16]. We extracted all the comments made by the developers from 1,860 projects using GitHub API. To get a reasonable estimate of personality profile, the *IBM Watson Personality Insights* recommends to have at least 600 words and states that accuracy increases as the number

[1] https://bit.ly/2sJbmgD

of words increases. Hence, we only considered those profiles who have left comments on GitHub and have more than 1,000 words across those comments. We applied the following preprocessing steps before we sent the text to the *IBM Watson Personality Insights* that includes:

- Removing all profiles whose communication language was not English. We used polyglot [2] library's language detector function to detect the dominant language in the text. Manually analyzing the comments revealed that developers sometimes used their native language to communicate with other fellow developers from the same country. Although the service provides personality scores for a few other languages apart from English, we wanted our scores to be consistent, and thus only chose English comments.
- We converted the comments available in the markdown format to html and removed the tags corresponding to the code. Comments made by the developers often have code included in the text. Since we only wanted natural language in the comments, we tried removing code from the text as much as possible.
- Lower-casing all the characters.
- Removing all the special characters except punctuation.

Once we send the raw text to the *IBM Watson Personality Insights*, this service returns a JSON object containing the percentile value between 0 and 1 for each personality trait that represents where the user stands relative to others. We gathered the personality traits of 29,396 developers. Additionally, we considered the pull requests only if we had both closer's and requester's extracted personality traits. This meant some of the developers' personalities were never considered because of not having a corresponding closer's or requester's personality information. Our final dataset included a total of 501,327 pull requests from 1,860 projects and had 16,935 developers. We describe the data modeling process in the next section.

3.1.3 Overview of the Data Modelling Process

Feature Selection

We used technical-, social-, and personality-related factors as independent variables and pull request acceptance as the binary dependent variable. Table 3.1 shows the different fea-

[2] https://pypi.org/project/polyglot/

Table 3.1: different features used with descriptive statistics

Category	Variables	Description	5%	Mean	Median	95%
Social Factors (Tsay et al.)	*social_distance*	A binary variable that tells if the pull request author follows the closer.	-	-	-	-
	prior_interaction	Indicates previous interaction of the requester in the project.	5	1840	640	7423
	followers	Total number of followers of the requester at the time of data collection. This is constant for all appearances of a specific requester in the data.	0	114	21	370
Technical Factors (Tsay et al.)	*test_file_present*	A binary variable that represents if the pull request contains test files.	-	-	-	-
	total_churn	Represents the total number of lines changed in a pull request.	0	1101	27	1959
	files_changed	Total number of file changed in a pull request	0	9.5	2	32
	num_comments	Total number of comments pertaining to the pull request	0	4	1	18
	main_team_member	Indicates if the requester is a core team member or not.	-	-	-	-
	team_size	Size of the core team of the project at the time of data collection.This is constant for all the pull requests of a particular project.	13	88	102	238
	stars	Indicates the number of stars in a project at the time of data collection.	3	3122	614	14135
	project_age	Age of project in days at the time of data collection	1126	1998	2019	2893
Personality Factors (Ours)	*openness*	Represents the openness of a requster/closer.	0.84	0.95	0.98	0.99
	conscientiousness	Represents the conscientiousness of a requester/closer.	0.11	0.36	0.34	0.67
	extraversion	Represents the extraversion of a requester/closer.	0	0.06	0.04	0.21
	agreeableness	Represents the agreeableness of a requester/closer.	0	0.02	0.0	0.7
	neuroticism	Represents the neuroticism of a requester/closer.	0.54	0.76	0.78	0.92
	diff_X_abs	Absolute difference between closer's and requester's personality.Where X is different personality traits.	-	-	-	-

tures used in the modelling process and also the descriptive statistics of pre-transformation values. The statistics include feature values at 5th percentile, 50th percentile (median), 95th percentile, and the mean.

We used social and technical factors (Table 3.1) in all research questions: RQ0–RQ3. We used the Big Five personality traits of the requester for RQ1, closer's personality traits for RQ2, and absolute difference in the personality traits between requester and closer for RQ3.

Data Preparation

We examined the distribution of the data and found that several features were skewed. Thus, they were normalized via a log transformation. Since the range of the each feature varies drastically, we scaled the data using the default `scale()` function provided by *R* [77]. This function transforms the features using the z-score transformation which subtracts the mean and divides by the standard deviation so that each feature has a unit standard deviation. Furthermore, to remove correlated features, we used variable clustering on the features and used $|\rho| = 0.6$ as the threshold. Variable clustering analysis uses hierarchical clustering on the correlation values derived from either a Pearson or a Spearman correlation test, to group features together. We used Spearman correlation due to its robustness against non-normal data [85]. We found that `file_changes` was grouped together with `total_churn` and thus decided to remove `file_changes`. After removing, we computed the clusters again but did not find any more correlations. We further analyzed the case of multicollinearity in the data by using Variance Inflation Factor (VIF) and setting the threshold as 5 to identify them. We did not identify any multicollinearity in the data.

Model Construction

We used a mixed effects logistic regression model (the same model used by Tsay et al.) provided by `lme4` library in *R* [13]. Pull requests grouped into a hierarchy of repository names, requester and closer, and represent data over time. By using a mixed effects model, we explicitly model the correlation among the hierarchy. We used the project name, requester and closer as random effects, and used features described in Table 3.1 as fixed effects. We report the odds ratio of the mixed effects model and also the confidence interval of the odds ratio using bootstrapping. Bootstrapping is a method to estimate the empirical distribution by generating data multiple times with replacement. A bootstrap sample is a resample of the same size, but with replacements. If u is a true distribution of the statistic, we aim to find $u*$, which is an approximation of the the true distribution. We retrieved 50 bootstrap samples from the dataset and separately modelled each of them. Later, we extracted the odds ratio from each of the 50 models and computed confidence interval on them. For each research question, we report the influence of different features as an increase or a decrease in odds ratio on pull request acceptance. If the odds ratio is greater than 1, this represents a positive relationship of the independent variable on the dependent variable, whereas a value less than 1 suggests negative relationship. Lastly, we consider the odds ratio to be significant only if the p-value is less than 0.001.

3.2 Results

For each research question, we will address our motivation, approach, and results. We will follow up the results with explanations from the psychology literature in the Discussion.

3.2.1 (RQ0) Can Tsay et al.'s results be replicated on a more recent dataset?

Motivation: We replicated Tsay et al.'s [87] findings on a more recently mined dataset to determine if the results still hold. GitHub has experienced a tremendous growth in the past five years with the number of repositories increasing from 4.6 million in 2013 to more than 96 million in 2018 [5]. During the past five years, new and advanced tools have been developed to facilitate better project management and to track code changes precisely. The replication process provides an important insight on the generalizability of Tsay et al.'s results to a dataset extracted at a different point in time. Additionally, by creating a baseline model with only technical and social factors helped us in comparing other personality models with social, technical, and personality factors.

Approach: We used the same modelling technique as suggested by Tsay et al. [87] on a new set of data. We used a mixed effects logistic regression using only the features used in their work.

Results: We replicated the effects of social and technical factors on pull request acceptance. All factors had similar overall influences except for followers count, for which we did not find significant effects. We found the effects of having test file included in a pull request decreased from 17 percent (reported by Tsay et al. [87]) to 8 percent in our model. Social distance and prior interaction were still the most important factors, influencing the pull request acceptance positively. Number of comments in a pull request and the number of stars of a project were the most important factors that influence pull request acceptance negatively. Although there were fluctuations in the odds ratio of all the features, we believe Tsay et al.'s overall results still stand. The change in some of the factors may be due to the selection of projects. Since we only included projects that have more than 250 closed or merged pull requests, it may have resulted in selection of projects that have been active for a longer duration. Table 3.2 provides a comparison between our results and Tsay et al.'s results.

3.2.2 (RQ1) Does the personality of a requester affect the likelihood of the pull request being accepted?

Motivation: A requester can either be a core team developer or an outside contributor. They often make useful suggestions in the form of pull request that can have a positive impact on the project. Previous work by Marlow et al. [54] suggests that project owners use personality clues derived from the communication activities of a developer to get an idea of what the person is like to work with. By examining the requesters, we want to determine whether having specific personality traits lead to a higher likelihood of pull request acceptance.

Approach: We modelled the personality traits of the requester—Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism—along with the features used by Tsay et al. [87]. Similar to RQ0, we used a mixed effects logistic regression to model the data.

Results: We observed the following effects (Table 3.3).

Openness: Openness of a requester was positively associated with pull request acceptance, increasing acceptance likelihood by 7 percent per unit increase.

Conscientiousness: Conscientiousness of a requester was positively associated with pull request acceptance, increasing acceptance likelihood by 6 percent per unit increase.

Extraversion: Extraversion of a requester was negatively associated with pull request acceptance, decreasing acceptance likelihood by 6 percent per unit increase.

Agreeableness: Agreeableness of a requester was positively associated with pull request acceptance, increasing acceptance likelihood by 2 percent per unit increase but the result was not significant.

Neuroticism: Neuroticism of a requester was negatively associated with pull request acceptance, increasing the acceptance likelihood by 3 percent per unit increase but the result was not significant.

Openness and Conscientiousness have positive effects on the pull request acceptance, while Extraversion has a negative effect.

Table 3.2: odds ratio of the model for RQ0

Variables	**Ours**	**Tsay et al's**
(Intercept)	2.81	1.106
test_file	1.08 (***)	1.171 (***)
total_churn	0.9 (***)	0.738 (***)
files_changed	-	0.927 (***)
social_distance	2.35 (***)	2.870 (***)
num_comments	0.68 (***)	0.454 (***)
prior_interaction	1.53 (***)	1.356 (***)
followers_current	1.	1.181 (***)
main_team_member	1.16 (***)	1.636 (***)
age_current	0.91 (***)	0.820 (***)
team_size	0.99	0.954 (**)
stars_current	0.53 (***)	0.648 (***)
test_file:num_comments	1.12 (***)	1.106 (***)
total_churn:num_comments	1.06 (***)	1.169 (***)
file_changed:num_comments	-	1.035 (***)
social_connection:num_comments	0.92 (***)	0.796 (***)
num_comments:prior_interaction	1.05 (***)	1.142 (***)
AIC	**394718**	-

*$p < .05$. **$p < .01$. ***$p < .001$.

3.2.3 (RQ2) Does the personality of a closer affect the likelihood of the pull request being accepted?

Motivation: A closer, unlike a requester, is always part of the core team. Their responsibilities include reviewing pull requests, suggesting changes, prioritizing tasks, and engaging in secondary functions such as attracting new developers. They are essential part of the pull request process as their decisions directly impact the functioning of the project and can move the project forward. Wang [90] conducted an empirical study on 116 projects and found positive evidence of project manager's personality affecting project success. While some closer may choose to reject an imperfect pull request, others may be more supportive and try to address the imperfection with the requester. By anaylzing the closers, we aim to understand whether specific personality traits affect the likelihood of the pull request getting accepted.

Approach: We replaced the requester's personality traits with the closer's personality traits and modelled them along with the factors used in RQ0.

Results: We observed the following effects (Table 3.4).

Openness: Openness of a closer was positively associated with pull request acceptance, increasing acceptance likelihood by 5 percent per unit increase but the result was not significant

Conscientiousness: Conscientiousness of a closer is positively associated with pull request acceptance, increasing acceptance likelihood by 12 percent per unit increase.

Extraversion: Extraversion of a closer was positively associated with pull request acceptance, increasing acceptance likelihood by 6 percent per unit increase.

Agreeableness: Agreeableness of a closer was positively associated with pull request acceptance, increasing acceptance likelihood by 1 percent per unit increase but the result was not significant.

Neuroticism: Neuroticism of a closer was positively associated with pull request acceptance, increasing acceptance likelihood by 8 percent per unit increase.

Conscientiousness, Extraversion, and Neuroticism have positive effects on the pull request acceptance.

Table 3.3: Odds ratio of the mixed effect model with requester's personality

	Single Run	**Bootstrapping**
Variables	**Odds Ratio**	**95% CI**
(Intercept)	2.87 (***)	-
test_file	1.08 (***)	[1.05,1.11]
total_churn	0.9 (***)	[0.89,0.9]
social_distance	2.35 (***)	[2.59,2.8]
num_comments	0.68 (***)	[0.65,0.69]
prior_interaction	1.52 (***)	[1.51,1.57]
followers_current	0.99	[0.95,0.98]
main_team_member	1.15 (***)	[1.12,1.2]
age_current	0.91	[0.83,0.93]
team_size	0.99	[0.85,0.98]
stars_current	0.54 (***)	[0.45,0.49]
openness	1.07 (***)	[1.05,1.08]
conscientiousness	1.05 (***)	[1.03,1.07]
extraversion	0.94 (***)	[0.93,0.95]
agreeableness	1.01	[1.,1.02]
neuroticism	0.97	[0.94,0.98]
test_file x num_comments	1.12 (***)	[1.11,1.15]
total_churn x num_comments	1.06 (***)	[1.06,1.08]
social_connection x num_comments	0.92 (***)	[0.9,0.96]
num_comments x prior_interaction	1.05 (***)	[1.06,1.08]
AIC	**394635**	

*$p < .05$. **$p < .01$. ***$p < .001$.

Table 3.4: Odds ratio of the mixed effect model with closer's personality

Variables	**Single Run**	**Bootstrapping**
	Odds Ratio	**95% CI**
Intercept	2.87	-
test_file	1.08 (***)	[1.05,1.11]
total_churn	0.90 (***)	[0.89, 0.91]
social_distance	2.35 (***)	[2.35,2.83]
num_comments	0.68 (***)	[0.65,0.68]
prior_interaction	1.49 (***)	[1.52,1.56]
followers_current	0.98	[0.94,0.99]
main_team_member	1.16 (***)	[1.12,1.21]
age_current	0.92 (***)	[0.86,0.95]
team_size	0.97	[0.88,1.0]
stars_current	0.54 (***)	[0.44,0.51]
openness	1.05 (**)	[1.02,1.10]
conscientiousness	1.12 (***)	[1.11,1.18]
extraversion	1.06 (***)	[1.06,1.13]
agreeableness	1.01	[0.99,1.04]
neuroticism	1.08 (***)	[1.06,1.14]
test_file x num_comments	1.12 (***)	[1.08,1.16]
total_churn x num_comments	1.06 (***)	[1.05,1.08]
social_connection x num_comments	0.92 (***)	[0.88,0.95]
num_comments x prior_interaction	1.05 (***)	[1.05,1.08]
AIC	**394548**	

*$p < .05$. **$p < .01$. ***$p < .001$.

3.2.4 (RQ3) Does a difference in personality between requester and closer affect the likelihood of the pull request being accepted?

Motivation: There have been numerous contradictory studies by software engineering community on the effects of personality differences in the offline setting. Baddoo and Hall [10] showed that having differences in personalities sometimes lead to personality clashes which demotivate and frustrate developers. Similarly, Karn et al. [49] observed occasional personality clashes in one of the software engineering teams they studied. Conversely, Carpretz and Ahmed [20] showed that having diverse personality is an essential part to solve different problems in a software development environment. A study by Chen et al. [22] looked at Wikipedia, an another online collaborative environment and found diversity in tenure and interests among the people had a positive effect on productivity. There has been no known study that examines the effect of personality difference in an online software engineering collaborative environment and hence motivated us to look into the effect of personality differences in pull request acceptance likelihood. Analyzing the differences in the personality traits will help us understand whether a higher difference leads to personality clashes or prompts more discussions, and in turn a higher pull request acceptance.

Approach: We considered the effects of personality differences in the model by adding the absolute differences between personality traits of requester and closer along with the features used in RQ0.

Results: We observed the following effects (Table 3.5).

Openness: The difference between requester's Openness and the closer's Openness was positively associated with pull request acceptance, increasing the acceptance likelihood by 1 percent per unit increase but the result was not significant.

Conscientiousness: The difference between requester's Conscientiousness and the closer's Conscientiousness was positively associated with pull request acceptance, increasing the acceptance likelihood by 29 percent per unit increase.

Extraversion: The difference between requester's Extraversion and closer's Extraversion was positively associated with pull request acceptance, increasing the acceptance likelihood by 12 percent per unit increase.

Agreeableness: The difference between requester's Agreeableness in closer's Agreeableness was associated with pull request acceptance, increasing the acceptance likelihood by 2 percent per unit increase. Although significant, the effect size is extremely small for the factor to be considered important.

Neuroticism: The difference between requester's Neuroticism and the closer's Neuroticism was positively associated with pull request acceptance, increasing the acceptance likelihood by 22 percent per unit increase.

The absolute differences in Conscientiousness, Extraversion, and Neuroticism affect the pull request acceptance positively.

A summary of results for different research questions is presented in Table 3.6. We also saw a decrease in the Akaike Information Criterion (AIC) statistic for the models used in RQ1–RQ3 from the model used in RQ0. This suggests that models with personality traits have a better fit compared to the model without personality traits. For example, in RQ3, the AIC decreased from 394,718 to 390,495 (Table 3.5 and 3.2).

3.3 Discussion

We now present additional insights gained by applying the literature in personality psychology.

Individuals with high Openness are open to new experiences and enjoy discussing new ideas. Since GitHub is a collaborative software environment, we hypothesized that both requester and closer would be highly open. Requesters would come up with new ideas and suggest novel features, while closers would be open to these new changes. As hypothesized, our result shows pull requests were more likely to be accepted when the requester was high on Openness. This is not surprising as Openness is also known as the creativity domain [60]. As such, we believe highly open requesters are likely to submit novel and interesting pull requests. Moreover, they tend to articulate their thoughts and ideas more cogently [31] and are therefore more persuasive to the project members. Given these characteristics, new suggestions and pull requests are more likely to be accepted. Similarly, when closers were high on Openness, pull requests were more likely to be accepted, but this result was not significant.

Conscientiousness being the measure of orderliness and dutifulness, we hypothesized that the effect on pull request acceptance would be positive regardless of the requester's identity (i.e., being a core team member vs. outside contributor). It is worth noting that the Conscientiousness of the requester was not as significant as other personality traits; nevertheless, the results were as expected. Individuals who are high on Conscientiousness are well-organized and have strong work ethics [25]. This implies that highly conscientious

Table 3.5: Odds ratio of the mixed effect model with personality differences

Variables	**Single Run**	**Bootstrapping**
	Odds Ratio	**95% CI**
(Intercept)	3.34 (***)	-
test_file	1.09 (***)	[1.05,1.12]
total_churn	0.92 (***)	[0.9,0.93]
social_distance	1.81 (***)	[1.86, 2.03]
num_comments	0.66 (***)	[0.65,0.67]
prior_interaction	1.66 (***)	[1.63,1.69]
followers_current	1.07 (***)	[1.06,1.11]
main_team_member	1.27 (***)	[1.23,1.31]
age_current	0.92 (***)	[0.87,0.93]
team_size	0.96	[0.89,1]
stars_current	0.55 (***)	[0.44,0.50]
diff_openness_abs	1.01	[1.01,1.04]
diff_conscientiousness_abs	1.29 (***)	[1.29,1.35]
diff_extraversion_abs	1.12 (***)	[1.11,1.16]
diff_agreeableness_abs	1.02 (***)	[1.0,1.04]
diff_neuroticism_abs	1.22 (***)	[1.21,1.27]
test_file x num_comments	1.11 (***)	[1.09,1.15]
total_churn x num_comments	1.06 (***)	[1.05,1.07]
social_connection x num_comments	0.93 (***)	[0.89,0.97]
num_comments x prior_interaction	1.05 (***)	[1.05,1.08]
AIC	**390495**	

*$p < .05$. **$p < .01$. ***$p < .001$.

Table 3.6: Effect of personality traits on pull request acceptance for different research questions.

Research Question	Openess	Conscientiousness	Extraversion	Agreableness	Neuroticism
RQ1 (Requester's personality)	7%	5%	-6%	NS	NS
RQ2 (Closer's personality)	NS	12%	6%	NS	8%
RQ3 (Personality difference)	NS	29%	12%	2%	22%

NS implies Not Significant. Each cell represents if it increases or decreases the likelihood of pull request acceptance.

requesters are detail-oriented and are likely to make fewer errors in their work, thereby increasing their likelihood of acceptance.

Previous work noted that individuals with high Conscientiousness make effective managers and are likely to occupy leadership roles [12]. As such, we suspect that highly conscientious closers assume the leadership role, motivating others to produce good work. This is reflected in a sample comment (Table 3.7) where a highly conscientious developer said: "Thanks for reaching out. I want to loop in a few people..." (Table 3.7). As a result, if a higher quality of work is achieved, then it is more likely to be accepted by the closers.

Individuals with high Extraversion are gregarious and very active in social settings. We hypthesized that both requester's and closer's Extraversion would have a positive association with pull request acceptance. This was not true in the case of the requester; in particular, pull requests were less likely to be accepted when the requester was high on Extraversion. This may be explained by the assertiveness characteristic of highly extraverted individuals [57], which may be interpreted as aggressive. For the closer, pull requests were more likely to be accepted when they were high on Extraversion, as expected. Mehrabian, for example, found a positive correlation between Extraversion and dominance [62]. This may suggest that when the closer is more extraverted, their assertiveness is more likely to be interpreted as being a good leader. Our result is also consistent with Wang and Li's study that found a positive correlation between Extraversion and leadership performance of the project managers [90].

Agreeable individuals are thought to be compassionate and polite; as such, we hypothesized a positive effect on pull request acceptance regardless of the developers' identity. However, we did not find any significant effects of agreeableness associated with requesters or with closers.

While neuroticism of a requester did not have a significant effect on the likelihood of

Table 3.7: Sample comments from profiles with high scores on each trait

Pesonality	**Sample Comments**
Openness	*That's possible, though it wouldn't be my approach. I would go for a context_id integer within the context structure and set it via mbedtls_ssl_init() with an unique id. But in the end, it's you who will read the logfile when I have something to report. :) So, i'm okay to try this method.*
	Ok better than the previous commit but still not perfect :). Not fixed yet then, lets continue
Conscientiousness	*I'll only merge if all the checks are passed.*
	Thanks for reaching out. I want to loop in a few people. Can you let me know if this is consistent for all web views or only initial ones? I want to figure out if this is because the screenshot is happening before the view loads or if it's the way the screenshot is taken.Thanks!
Extraversion	*Its okay if it didn't work, thanks for attempting though. Keep the contributions coming in!!!*
	Cheers, this seems like a big contribution. Thanks for the pull request. This will improve the app's functionality drastically.
Agreeableness	*I apologize for troubling you. I don't intend to add this package against your will.*
	Thank you for pointing out it. I have amended the pull request
Neuroticism	*I'm not an accountant but I have worked more than you in this field.I'm pretty sure that US based businesses need to charge VAT for EU customers. Digital goods are actually considered services. Do it like WooCommerce to solve this tax nightmare.*
	Thats a bummer, anyway if its not fixed soon i'll revert the master to stable.

pull request acceptance, pull requests were more likely to be accepted when the closer was high on Neuroticism. One of the reasons for this association may be due to the fact that closers express more negative sentiments in the communication thread to help the requesters fix their pull requests. If a requester, for example, suggests new features, the closer with high Neuroticism may share their opinions as to how the new feature might not work. While being more critical, the closers give better feedback and hence get the pull request to a better condition. Further studies are required to understand the effects of high Neuroticism on pull request acceptance.

When examining the absolute difference between the requester and the closer, we found that more pull requests were more likely to be accepted when the difference was greater. This greater difference or diversity argument has been explored in many fields including artificial intelligence and organizational behavior. When forming multi-agent teams, a greater strategic diversity, even with weak agents, outperformed strong but less diverse teams [53]. In addition, a comprehensive review noted that diversity of personality in the workplace led to positive performance [52]. Following this evidence, we believe that the diverse pool of developers in a given project is associated with higher likelihood of pull request acceptance.

While the effects of personality traits on pull request acceptance are present, we see that non-personality factors have more effects on the pull request acceptance. In all the research question, we see social distance to have the highest importance. For RQ1, the most important personality factor was openness (1.05 percent) compared to social distance (1.25 percent). Neuroticism had the highest importance (1.12 percent) among the personality factors, but social distance was the most important overall (2.22 percent). Finally for RQ3, difference in conscientiousness (1.3 percent) came out as the most important but prior interactions (1.66 percent) was the most important overall. It is evident that social factors are the most influential factors when it comes to predicting pull request acceptance. In addition to social factors, both personality factors and technical factors are important in the pull request acceptance predictions. Our results specifically show that personality factors were as important as the technical factors. In the model with the requester's personality, we have Openness that has similar odds ratio to test file inclusion and number of line changed. Further, in the model with closer's personality, we have Conscientiousness and Neuroticism that are better predictors than technical factors.

3.4 Threats to Validity

Recall that the personality traits of developers were obtained using the *IBM Watson Personality Insights*. It is possible that the personality model generated by this service is not actually representative of the developers' true personality. Even if this is true, we claim that this is not an issue as we only care about the digital footprint or perceived personality—that is, how developers are perceived by other developers on GitHub. Moreover, there is evidence that self-reported personality measures are not the only valid way to measure one's personality. Self-reports show strong correlations with observer ratings [61]. Future works could explore indirect communication media, such as Gitter [3], Discord [4], or Slack channels [5], to extract personality traits and remap them to GitHub usernames. These communication channels are known to be more informal: developers often express sentiments more casually and use emojis in the conversations, making it easier to extract their digital personalities.

We retrieved the personalities of the developers from the comments on the pull request discussion thread. We agree there may be some concerns with the performance of the service on software engineering-specific text content, but we mitigate this issue by using different regular expressions to remove code from the text. Additionally, since we only consider developers who have communicated at least 1,000 words as comments in a GitHub project, some developers were ignored. We believe the 16,935 sample of highly active users is large enough to perform an empirical analysis.

There may be some concerns with the data not being representative of the true population. We tackled this concern by taking project samples from 2 different sources: projects derived from the original Tsay et al.'s work [87] and a sample extracted from the set of valid projects from RepoRepear. We analyzed a total of 501,327 pull requests with 16,935 users, which in itself is a good representation of the population. Further, we followed the same process as used by Tsay et al., and additionally perform bootstrapping and report the confidence intervals to make our results more reliable.

Finally, in this study we have only considered the requester and closer while evaluating the effects of personality factors. There are other developers who might express their opinions in the pull request thread who might have valuable feedback for the requester, but we believe the final decision is still made by the closer and thus chose to solely focus on closer's personality rather than personality of the group.

[3] https://gitter.im/
[4] https://discordapp.com
[5] https://slack.com/

To summarise, in this chapter, we investigated the effects of five personality traits on pull request acceptance. We looked at the personality of the requester, closer and the personality differences between the requester and the closer. We found that higher openness of requester was positively associated with pull request acceptance. A higher conscientiousness from both the requester and closer resulted in a better pull request acceptance. A higher extraversion of the requester was associated negatively with pull request acceptance, whereas it was positively associated for the closer. Higher neuroticism was associated positively with pull request acceptance. Finally, the higher the difference in personality, we see a higher pull request acceptance.

Chapter 4

Analysing the effects of emotions of GitHub comments in pull request decisions.

Emotions are exhibited both while communicating on an online medium or while interacting with a person offline. Developers collaborating online, need to be aware of the emotional state of the pull request and respond appropriately. Understanding the emotional state of the discussion will help the developers to get a better idea of the expectations and react suitably. Hence, to investigate the influence of emotions on pull request decisions, we conduct this experiment. In this chapter, we first discuss the experiment methodology, explain the motivation, and formulate five research questions. Later, we summarize the results and discuss the insights gained from the results by outlining possible causes.

4.1 Methodology

In this section, we describe our data selection, data extraction and data modelling process.

4.1.1 Data Selection

From a set of 24 million pull request collected, pull requests who had at least five comments were selected. This filtering criterion removed pull requests that did not have enough communication among the developers in a pull request thread. For each pull request, we

retrieved all the comments. We found that developers mostly communicated either via pull requests comments or review comments. Additionally, a manual examination of 30 randomly selected commit comments revealed code specific information. We observed the commit comments in most of the cases expressed neutral emotions. In a commit comment, team managers tend to ask code-specific questions such as "what does this code do?" or "can you optimise this block?" and hence we chose to exclude the commit comments. We retrieved a total of 2.2 million comments across 406,600 pull requests with their information about the comment author's association in the project. We used 406,600 pull requests for evaluating the RQ1 but only used a subset for the remaining questions. For RQ2 and RQ3 we used 327,442 pull requests as we removed all those pull requests where the requester has not commented a single time. Similarly, for RQ4 and RQ5, we used 335,358 pull requests that included at least one comment from the closer. We classified the comments retrieved from the Github API as having negative, neutral, or positive with our state of the art model.

4.1.2 Classification of Comments to emotions

This section describes the process involved in classifying software engineering comments to emotions. First, we describe the training process, followed by experiments on standard datasets, and finally, we evaluate the performance of the model on Github comments.

UlmFit on StackOverflow

A language model trained on a general English based resources does not perform equally on software engineering data. We developed a language model using software engineering specific data to help the model learn different linguistic variations in software engineering. We describe the data selection process as follows:

- We used the Stackoverflow dump available at Archive.org [1].
- We extracted 42,850,540 posts from the dump.
- We then randomly selected 4.2 million posts from the extracted dump.
- We removed all the code from the posts using different regex pattern provided by Baltes et al. [11].

- We Tokenised and Vectorised the sentences in the post. We used spacy library [41] to tokenise the sentences and for vectorising we used the function provided by FastAI library. [42]. We also restricted our vocabulary to only include the most common 50,000 words.
- We used an implementation of AWD LSTM provided by FastAI library. [42] for training the language model. A total of 2 language model were trained: one forward and one backward.
- We did not use any pre-trained word embedding. The word embedding was trained together with the language model.
- Once the language model was trained, we applied the transfer learning technique provided by UlmFit for specific emotion recognition tasks. More details is described in the next subsection.

Parameters for the language model: We used a batch size of 128 for training both the forward and backward language models. The language model architecture consists of AWD LSTM with Embedding size 400, 3 Layers of Weight Dropped LSTM with sizes 1150, 1150, and 400 respectively and a final softmax layer of the size of the vocabulary. We used an embedding dropout probability was 0.02 and set the input dropout probability to 0.25. The weight dropout probability was 0.2, the hidden output dropout was 0.15, and the final output dropout was 0.1. We use the max sequence size of 70 words for both the models. Using the same parameters for both the language models, we trained the models for 30 epochs each.

Fine Tuning: Once both the language models finished training, we used UlmFit to transfer these language model on to the downstream task of emotion classification. As previously described in the Section 2.7, we first adapt our language model to the task-specific dataset by fine-tuning it for 5 epochs. We then replace the final softmax layer with the task-specific layer (in our case, the number of classes in the data). Finally, we use gradual unfreezing to fine-tune the language model. We train only 3 layers through our model.

- We first train just the softmax (1st layer) layer for 5 epochs.
- Then we unfreeze the 1st LSTM (2nd layer) layer from the last and train 1st and 2nd layer together for 10 epochs.
- Then we unfreeze the 2nd LSTM layer (3rd layer) from the last and train 1st, 2nd and 3rd layer together for 10 epochs.

Dataset	Negative	Positive	Neutral
JIRA	636	290	-
Gerrit [1]	398	1202	-
SOSentiments	1202	1527	1694
SOJava	178	131	1191

Table 4.1: Number of data points for different labels in each dataset.

To restrict our model over-fitting, we stopped training the model after unfreezing the first 3 layers. The learning rate was picked after manual inspection of the model's performance and is different for each emotion dataset. We use Stratified split cross-validations to evaluate the overall performance of the model. The F1 score of the model on each dataset is measured for each fold separately. Finally, we present the average metrics aggregated from all the folds.

Experiments with Standard Datasets

We evaluate our model on four different datasets. We perform stratified cross-validation with 10 splits similar as used by Ram et al. [79]. We use this paper as a reference to compare our model's performance to other methods:

JIRA: This dataset was curated by Ortu et al. [71]. The dataset consists of issue comments labelled into love,sad,joy and anger. For the experiment we only consider the emotion polarity and hence we reduce the labels into positive (love and joy) and negative (sad and anger). It has 636 negative comments and 290 positive comments.

Gerrit: This dataset was curated by Ahmed et al. [8] for developing their tool SentiCR. The dataset consists code reviews labelled as positive, neutral and negative. Similar to Ram et al.we also use their approach to split the dataset into negative and non-negative classes to overcome class imbalance. The dataset has 398 negative comments and 1202 non-negative comments.

Stackoverflow Java lib: This dataset was curated by Lin et al. [51] for understanding java libraries' sentiments and using the sentiments for recommendation. The dataset is labelled as positive, negative and neutral. This dataset includes 178 negative, 141 positive and 1191 neutral comments.

Stackoverflow Sentiments: This dataset was curated by Calefato et al. [18] for developing the tool Senti4SD. The authors provide a pre split training and testing test for

easy comparison. We used the model trained on this dataset to classify Github comments. We combined the training and testing set and then used stratified cross validation on the combined dataset. The dataset contains 1202 negative, 1527 positive and 1694 neutral comments.

We outperform the previous methods on all the standard software engineering datasets. Table 4.2 compares the result of different tools on the above datasets. We improve the performance on the Gerrit dataset by increasing the F1-score by 4 percent, on StackOverflow sentiments dataset we increase the F1-score by 3 percent, on StackOverflow java library dataset we increase the F1-score by 12 percent and on JIRA dataset we match the previous best.

Results on Github comments

Due to the lack of quality annotated Github comments, we used the model trained on the Stackoverflow sentiments [18] to classify Github comments. Manual inspection of Github comments and Stackoverflow found similar word uses and linguistic pattern. Additionally, the dataset is adequately balanced, which helps the models to learn better decision boundary. We classified all of the 2.2 million Github comments into the respective sentiment category. The classification process took 30 mins on NVIDIA Tesla P100 GPU. To verify the accuracy of the model, we randomly selected 108 comments with 36 from each class: negative, neutral, and positive. We divide this set into 2 groups of 54 comments each and ask 4 graduate students to annotate the comments. We provide the Cohen's Kappa score, which is used to measure the inter-annotator agreement. Additionally, we also provide the percentage of annotator agreement with the model.

Two different annotators annotated a set of 54 comments. Figure 4.1 shows the confusion matrices for different annotation and model combinations. We also computed the weighted Cohen's kappa [23] to measure the rater's agreement. We use a linear weighting scheme to compute the score due to emotion polarity following a linear ordering. The weighting scheme helps to penalize positive - negative disagreement more than positive - neutral disagreement. Table 4.3 shows the pairwise agreement scores for set 1. The inter-human agreement stands at 0.51, and human-model agreement is 0.45 and 0.55. The model and annotator have similar performances on the agreement.

Figure 4.2 shows the confusion matrices for annotators and the model labels for set 2. Table 4.4 shows that for the second comment set, the model performs slightly worse than human raters. The inter-human agreement stands at 0.58, whereas the human-model have

Dataset	Classifier	Negative	Positive/NN	Neutral	Avg F1
JIRA	SentiStrength	0.82	0.75	-	0.78
	SentiStrengthSE	0.82	0.75	-	0.78
	Senti4SD	0.91	0.92	-	0.92
	SentiCR	0.96	0.91	-	0.94
	CNN-LSTM	0.98	0.95	-	0.97
	UlmFit(Ours)	**0.98**	**0.95**	-	**0.97**
Gerrit *	SentiStrength	0.82	0.75	-	0.78
	SentiStrengthSE	0.39	0.86	-	0.62
	Senti4SD	0.59	0.89	-	0.74
	SentiCR	0.63	0.87	-	0.75
	CNN-LSTM	0.57	0.89	-	0.73
	UlmFit(Ours)	**0.72**	**0.91**	-	**0.82**
StackOverflow Sentiments	SentiStrength	0.78	0.90	0.75	0.81
	SentiStrengthSE	0.75	0.86	0.75	0.79
	Senti4SD	0.82	0.91	0.81	0.85
	SentiCR	0.76	0.90	0.80	0.82
	CNN-LSTM	0.82	0.91	0.83	0.85
	UlmFit(Ours)	**0.85**	**0.94**	**0.85**	**0.88**
StackOverflow Java Lib	SentiStrength	0.41	0.26	0.81	0.49
	SentiStrengthSE	0.26	0.26	0.87	0.46
	Senti4SD	0.43	0.26	0.90	0.53
	SentiCR	0.57	0.41	0.88	0.62
	CNN-LSTM	0.28	0.11	0.90	0.43
	UlmFit(Ours)	**0.72**	**0.57**	**0.93**	**0.74**

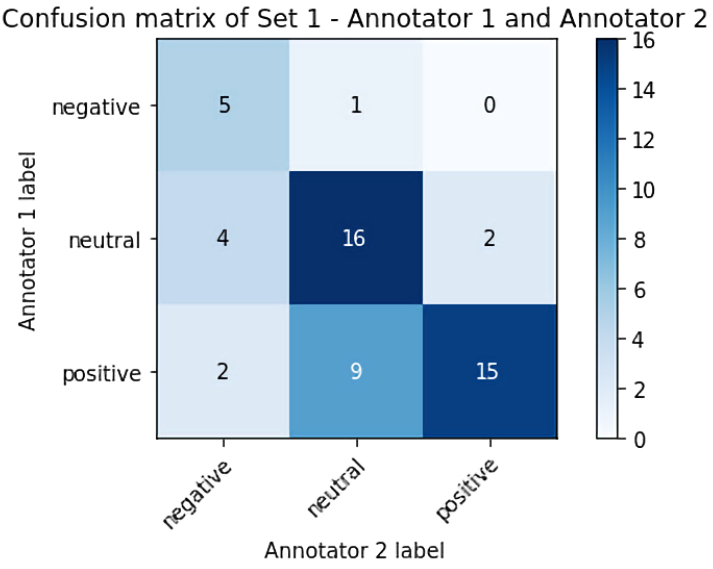
Table 4.2: Class wise F1 score performance of different tools on standarad SE datasets. * represents there is no positive class instead Non-negative class is used..

	Annotator 1	Annotator 2	Model
Annotator 1	-	0.51	0.45
Annotator 2	0.51	-	0.55
Model	0.45	0.55	-

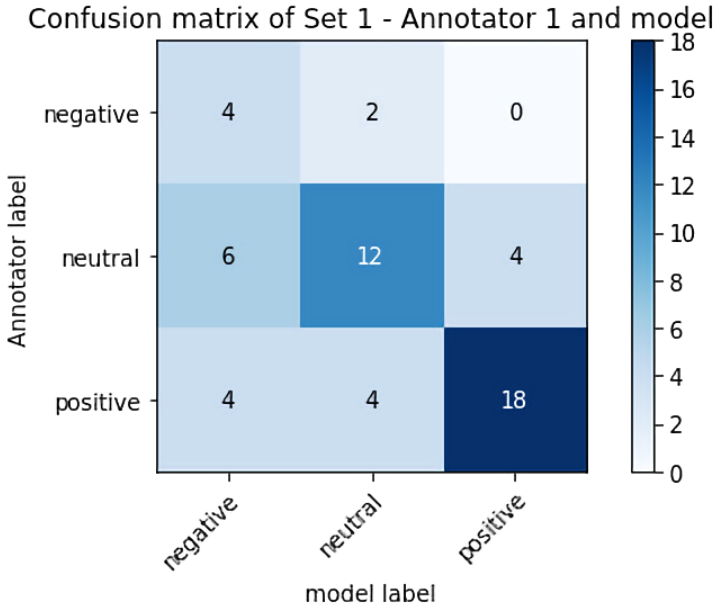
Table 4.3: Weighted Cohen's Kappa Score to measure pairwise agreement for set 1

	Annotator 1	Annotator 2	Model
Annotator 1	-	0.58	0.45
Annotator 2	0.58	-	0.43
Model	0.45	0.43	-

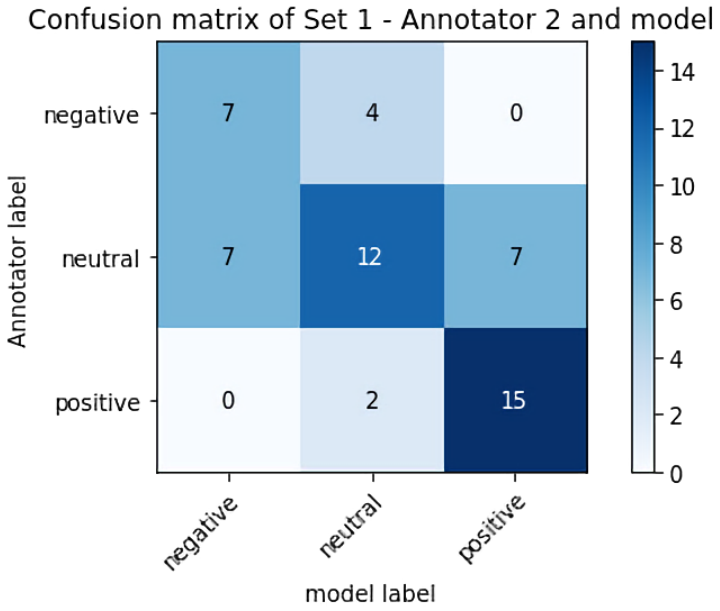
Table 4.4: Weighted Cohen's Kappa Score to measure pairwise agreement for set 2



(a) Confusion Matrix of Annotator response

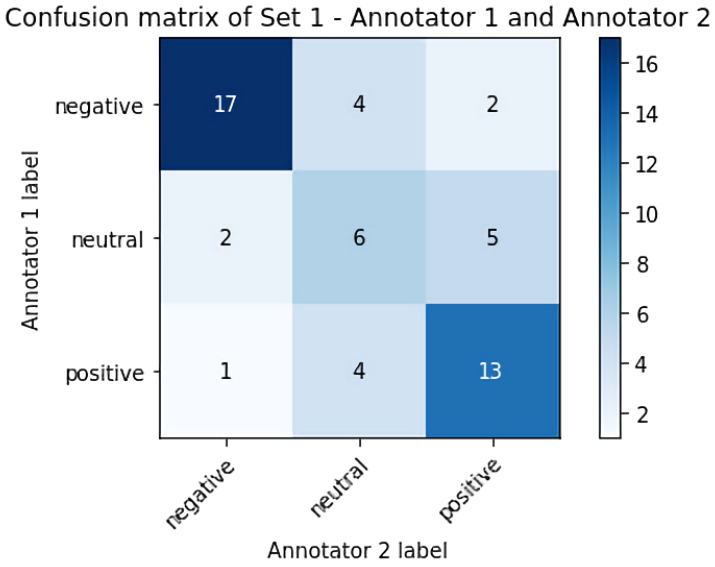


(b) Confusion Matrix of Annotator 1 and model

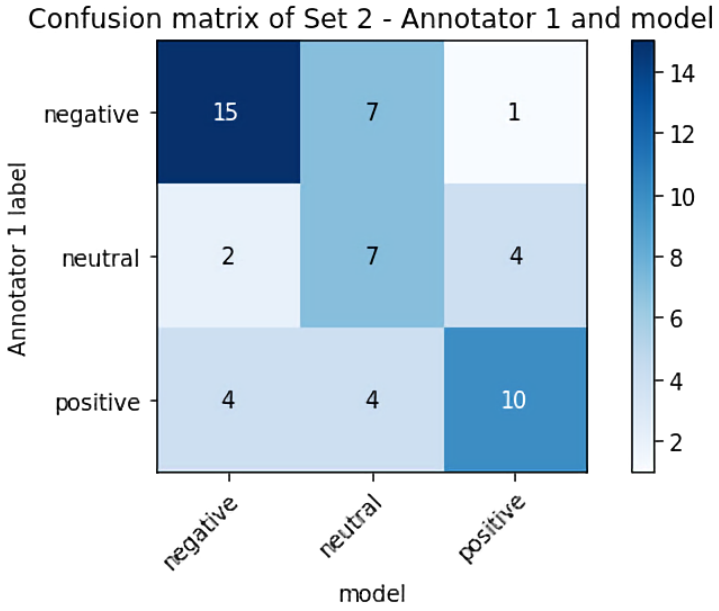


(c) Confusion Matrix of Annotator 2 and model

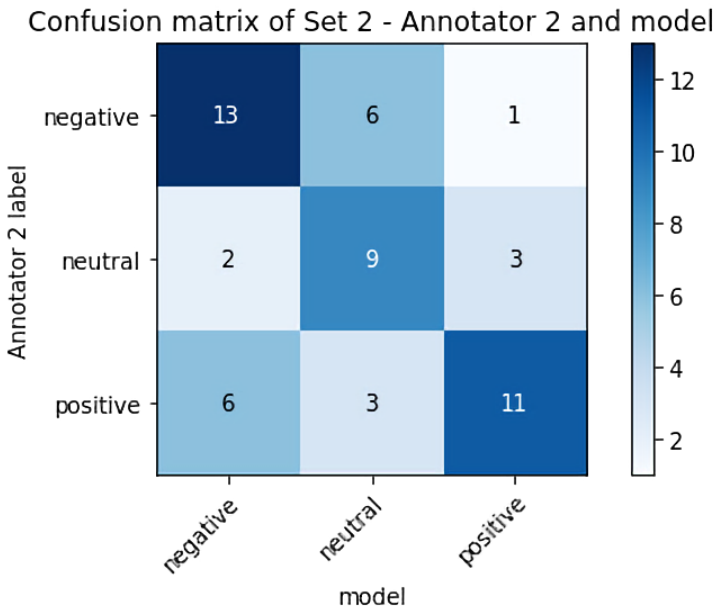
Figure 4.1: Confusion matrices for 54 comments in set 1



(a) Confusion Matrix of Annotator response



(b) Confusion Matrix of Annotator 1 and model



(c) Confusion Matrix of Annotator 2 and model

Figure 4.2: Confusion matrices for Set 2

a Kappa score of 0.43 and 0.45, respectively. The model agrees with at least one of the annotators 41 times in both the set of 54 comments resulting in an accuracy of 76 percent.

Finally, we grouped all the comments and derived many summaries described as follows.

- Percentage of positive, negative and neutral comments associated with each pull request. (Used in RQ1)
- Percentage of percentage of positive, negative and neutral comments of just the requester in each pull request. (Used in RQ2)
- Polarity of the emotion associated with the first comment of the requester in the pull request. (Used in RQ3)
- Percentage of positive, negative and neutral comments of the closer associated with pull request. (Used in RQ4)
- Polarity of emotion associated with the first response of the closer in a pull request was extracted. (Used in RQ5)

4.1.3 Overview of Data Modelling process

Feature Selection

Similar to the previous experiment, we used technical and social factors as base independent variables. We added different emotion summaries as features based on the research questions. The table 4.5 shows different features used in this experiment.

We used social and technical factors in all the research questions from RQ1 - RQ5. We used the percentage of negative, neutral and positive comments in a pull request for RQ1, the percentage of requester's positive, negative and neutral comments for RQ2, the percentage of closer's positive, negative and neutral comments for RQ4, the polarity of the requester's first response for RQ4 and the polarity of the closer's first response for RQ5.

Data Preparation

Similar to the previous experiment, we did a log transformation and scaled all the continuous features. We applied scaling to the emotion features used in RQ1, RQ2, and RQ4,

Table 4.5: different features used with descriptive statistics

Category	Variables	Description	5%	Mean	Median	95%
Emotional Factors (Ours)	*% of negative emotion*	Represents the percent of negative comments.	0.00	0.14	0.09	0.5
	% of positive emotion	Represents the percent of a positive comments.	0.00	0.26	0.21	0.80
	% of negative emotion(requester)	Represents the percent of requester's negative comments.	0	0.17	0.0	0.75
	% of positive emotion(requester)	Represents the percent of requester's positive comments.	0	0.21	0.0	1
	% of negative emotion(closer)	Represents the percent of closer's negative comments.	0	0.13	0.0	0.66
	% of positive emotion(closer)	Represents the percent of closer's positive comments.	0	0.3	0.2	1
	Emotion of First response (requester)	Represents the emotion of requester's first comment.	-1	0.1	0.0	1
	Emotion of First response (closer)	Represents the emotion of closer's first comment.	-1	0.13	0	1

i.e., the percentage of negative and positive comments. Additionally, for the emotion feature used in RQ3 and RQ5, we labeled negative as -1, neutral as 0 and positive as 1. We converted the categorical variables into ordinal values to evaluate the effect on pull request acceptance when the emotion increases towards positive. Finally, we checked for correlated variables with the help of Variable Clustering analysis. We used Spearman correlation with $|\rho| = 0.6$ as the threshold to identify correlated variables. We found that the percentage of neutral comments had a high correlation with the percentage of positive comments, and hence, we removed this variable. Additionally, we also used the Variance Inflation Factor with a threshold of 5 to remove multi-collinearity. We did not identify any multi-collinearity amongst the data.

Model Construction

We used the mixed-effects logistic regression model, as used in the previous experiment, with repository names and requester as random effects. We report the odds ratio from the model. Similar to all the models, we use a p-value of 0.001 to judge the significance of the result.

4.2 Results

For each research question, we explain the motivation, approach, and results. An analysis and explanation of the results are explained in the Discussion section of this chapter.

4.2.1 (RQ1) - Does the percentage of positive and negative emotions influence pull request acceptance?

Motivation: Past research has found that developers express emotions while communicating and also experience these emotions while reading other's comments [65, 76, 37, 54]. Since the pull request evaluation process entails discussions that express emotions, we aimed to measure the effect of having a higher percentage of positive and negative comments in the discussion thread on pull request decisions.

Approach: For this research question, on top of the base features (technical and social), we add 2 new independent variables, namely: the percentage of negative and positive comments. We used the mixed-effect logistic regression for modeling.

Results We observed the following effects (Table 4.6).

Percentage of Positive Comments: Unsurprisingly, we found that the percentage of positive comments contributed positively to the pull request acceptance. It is positively associated with pull request acceptance, increasing acceptance likelihood by more than 53 percent per unit increase.

Percentage of Negative Comments: We found that the percentage of negative comments also had significant effects. It is negatively associated with pull request acceptance, decreasing acceptance likelihood by 9 percent per unit increase.

Percentage of positive comments affects the pull request acceptance likelihood positively, whereas the percentage of negative comments affects negatively.

4.2.2 (RQ2) - Does the percentage of positive and negative emotions of the requester influence pull request acceptance and moderation effect caused by other social factors?

Motivation: In a pull request evaluation process, both the requester and the closer express emotions during the discussion. Project managers can raise questions, ask for more clarity,

Table 4.6: Odds ratio of the model for RQ1

Variables	**Odds Ratio**
Intercept	2.24
test_file	1.29 (***)
total_churn	0.98
social_distance	2.04 (***)
num_comments	1.02 (***)
prior_interaction	1.83 (***)
followers_current	1.01
main_team_member	1.13 (***)
age_current	0.92 (***)
team_size	0.99
stars_current	0.62 (***)
positive_perc	1.53 (***)
negative_perc	0.91 (***)
test_file x num_comments	0.98
total_churn x num_comments	1.01 (**)
social_connection x num_comments	0.96 (**)
num_comments x prior_interaction	0.95 (***)
AIC	**385254**

*$p < .05$. **$p < .01$. ***$p < .001$.

or suggest an alternative implementation to make the code better. We wanted to know if a requester responds to a project manager positively, does it increase the likelihood of pull request getting accepted. Exploring this research question will help us guide developers on how to respond and also understand what project managers expect from the requester. Additionally, we believe the effect of positive or negative comments is moderated by other social factors wherein the emotional effects get reduced if the requester is the principal team member. The effect can also reduce if the requester has a social connection to the manager or has a higher prior interaction. These effects are essential to consider as non-emotional factors play a stronger role in pull request evaluation. Project managers can weigh social factors to be more important and tend to ignore the emotional aspects of requester's comment due to the increased likelihood of trust.

Approach: For this research question, on top of the base features (technical and social), we add 2 new independent variables: the percentage of requester's negative and positive comments. We also add the interaction terms between social factors and both the emotion feature. We used a mixed effect logistic regression for modeling the data.

Results We observed the following effects (Table 4.7).

Percentage of Requester's Positive Comments: We found that the percentage of requester's positive comments contributed to the highest effect in the model. It is positively associated with pull request acceptance, increasing acceptance likelihood by 15 percent per unit increase.

Percentage of Requester's Negative Comments: We found that the percentage of negative comments also had significant effects. It is negatively associated with pull request acceptance, decreasing acceptance likelihood by 8 percent per unit increase.

Interaction effects with Social Distance: We found significant interaction effects between social distance and percentage of requester's positive comments, but the result was insignificant for the interaction between social distance and negative comments.

Interaction effects with Prior Interaction: We found significant interaction effects between both requester's positive and negative comments with prior interaction.

Interaction effects with requester being a team member: We did not find any significant interaction effects between both requester's positive and negative comments with main team member variable.

Percentage of requester's positive comments affects the pull request acceptance likelihood positively where as the percentage of requester's negative comments affects pull request acceptance negatively. We also found interaction effects of social distance and prior interaction on emotional factors to be significant.

Table 4.7: Odds ratio of the model for RQ2

Variables	**Odds Ratio**
Intercept	2.06
test_file	1.28 (***)
total_churn	1.0
social_distance	2.23 (***)
num_comments	1.02 (***)
prior_interaction	1.73 (***)
followers_current	1.01
main_team_member	1.09 (***)
age_current	0.94 (***)
team_size	0.92
stars_current	0.68 (***)
positive_perc	1.15 (***)
negative_perc	0.92 (***)
test_file x num_comments	0.97 (*)
total_churn x num_comments	1.01
social_connection x num_comments	0.93 (***)
num_comments x prior_interaction	0.97 (***)
social_connection x positive_percs	0.88 (***)
social_connection x negative_percs	0.96 (*)
main_team_member x positive_percs	0.99
main_team_member x negative_percs	0.99
prior_interaction x positive_percs	1.06 (***)
prior_interaction x negative_percs	1.02 (***)
AIC	**322527**

*$p < .05$. **$p < .01$. ***$p < .001$.

4.2.3 (RQ3) - Does the emotion of requesters first response aid to predict pull request acceptance and moderation effect caused by other social factors?

Motivation: In this research question, we wanted to know if the first response of the requester is predictive of pull request acceptance. A study by Marlow et al. [54] showed that project managers rely on the first impression of developers developed by seeing past work history, behaviors, and responses. The first impression plays a vital role in evaluating developers during edge cases scenarios. We hypothesize that more positive first response from the requester is better for pull request acceptance. Finally, we also analyze the interaction effects of different social factors on emotional effects.

Approach: For this research question, on top of the base features (technical and social), we add the emotion of the first response from the requester. We also add the interaction terms between the social factors and the emotion factor.

Results We observed the following effects (Table 4.8).

Emotion of first comment from the requester: We found that Emotion of first comment from the requester had a positive effect on the pull request acceptance increasing acceptance likelihood by 15 percent per unit increase.

Interaction effects: We find no interaction effects to be significant.

A positive first comment by requester is positively associated with pull request acceptance. There was no evidence of interaction effects of social factor on emotion of requester's first comment

4.2.4 (RQ4) - Does the percentage of positive and negative emotions of the closer influence pull request acceptance and moderation effect caused by other social factors?

Motivation: Similar to RQ2 of this experiment, instead of requester's emotion, we wanted to find if the emotion of closer influences the pull request acceptance. Project managers have the responsibility of taking their project forward. They evaluate the contributions and decide to accept or reject them. If the contribution is not in the right shape, they express negative emotions and suggest ways to improve the contribution. We believe, more the closer's a negative emotion, the lower the likelihood of acceptance. By asking this question, we want to confirm our understanding of emotions expressed by the closer.

Table 4.8: Odds ratio of the model for RQ3

Variables	**Odds Ratio**
Intercept	2.03
test_file	1.27 (***)
total_churn	1.00
social_distance	2.23 (***)
num_comments	1.02 (***)
prior_interaction	1.72 (***)
followers_current	0.99
main_team_member	1.09 (***)
age_current	0.94 (***)
team_size	0.91 (***)
stars_current	0.68 (***)
emotion	1.15 (***)
test_file x num_comments	0.97 (*)
total_churn x num_comments	1.0
social_connection x num_comments	0.95 (***)
num_comments x prior_interaction	0.98 (***)
prior_interaction x emotion	1.02
social_connection x emotion	0.95
main_team_member x emotion	0.99
AIC	**323464**

*$p < .05$. **$p < .01$. ***$p < .001$.

We also believe that the effect of closer's emotions in the pull request thread depends on other social factors. Factors such as social distance, prior interaction, and requester being the main team member moderate the effects of closer's emotion expressed in a pull request thread. For example, the harmful effect of having negative comments can vanish if the closer trusts the contributor.

Approach: For this research question, on top of the base features (technical and social), we add 2 new independent variables: the percentage of closer's negative and positive comments. Additionally, we also consider the interaction effects of social factors. We used the mixed-effect logistic regression for modeling.

Results We observed the following effects (Table 4.9).

Percentage of Closer's Positive Comments: We found that percentage of closer's positive comments had high effect size in the model. It is positively associated with pull request acceptance, increasing acceptance likelihood by 72 percent per unit increase.

Percentage of closer's Negative Comments: We found that percentage of negative comments also had significant effects. It is negatively associated with pull request acceptance, decreasing acceptance likelihood by 4 percent per unit increase.

Interaction effects with Social Distance: We found significant interaction effects between social distance and percentage of closer's positive comments but the result were insignificant for the interaction between social distance and negative comments.

Interaction effects with Prior Interaction: We did not find any significant interaction effects between both closer's positive and negative comments with prior interaction.

Interaction effects with requester being a team member: We found significant interaction effects between both closer's positive comments with main team member variable but did not find significant effects for negative comments.

Percentage of closer's positive comments affects the pull request acceptance likelihood positively where as the percentage of closer's negative comments affects pull request acceptance negatively. We also found interaction effects of social distance and requester being a team member on emotional factors to be significant.

Table 4.9: Odds ratio of the model for RQ4

Variables	**Odds Ratio**
Intercept	1.92
test_file	1.29 (***)
total_churn	1.0
social_distance	1.89 (***)
num_comments	1.04 (***)
prior_interaction	1.80 (***)
followers_current	1.01
main_team_member	1.14 (***)
age_current	0.95 (*)
team_size	0.95 (*)
stars_current	0.64 (***)
positive_perc	1.72 (***)
negative_perc	0.96 (***)
test_file x num_comments	0.97 (**)
total_churn x num_comments	1.01 (**)
social_connection x num_comments	0.95 (**)
num_comments x prior_interaction	0.96 (***)
social_connection x positive_percs	0.91 (***)
social_connection x negative_percs	1.05 (**)
main_team_member x positive_percs	0.90 (***)
main_team_member x negative_percs	0.97 (*)
prior_interaction x positive_percs	1.02 (**)
prior_interaction x negative_percs	1.01 (**)
AIC	**324950**

*$p < .05$. **$p < .01$. ***$p < .001$.

4.2.5 (RQ5) - Does the emotion of closer's first response aid to predict pull request acceptance and moderation effect caused by other social factors?

Motivation: In this research question, we hypothesize that just the first response of closer is predictive of pull request acceptance. We wanted to know if the closer makes the decision early in the pull requests. A first impression of the pull request can be expressed through the emotion of the first response of the closer. Additionally, the closer can respond less negatively initially to people he already trusts, has a social connection with requester or knows him personally if he is the main team member. To confirm this hypothesis, we consider the interaction effects of social factors. To understand this more we perform the following experiment.

Approach: For this research question, on top of the base features (technical and social) we add then emotion of closer's first response and its interaction terms with social factors. We used mixed effect logistic regression for modelling.

Results We observed the following effects (Table 4.10).

Emotion of first comment from the closer: We found that Emotion of first comment from the closer had a significant positive effect on the pull request acceptance increasing acceptance likelihood by 52 percent per unit increase.

Interaction effects with Social Distance: We found the effect of emotion decreases when the requester has a positive social distance.

Interaction effects with Prior Interaction: We did not find any significant effects for the interaction effects with prior interaction.

Interaction effects with requester being a team member: The effect of emotion decreases when the requester is a main team member.

A positive first comment by requester is positively associated with pull request acceptance. We found interaction effect of social distance and main team member.

Table 4.10: Odds ratio of the model for RQ5

Variables	**Odds Ratio**
Intercept	1.74
test_file	1.30 (***)
total_churn	1.00
social_distance	2.0 (***)
num_comments	1.03 (***)
prior_interaction	1.74 (***)
followers_current	0.99
main_team_member	1.12 (***)
age_current	0.95 (**)
team_size	0.91 (***)
stars_current	0.68 (***)
emotion	1.52 (***)
test_file x num_comments	0.97 (**)
total_churn x num_comments	1.01 (***)
social_connection x num_comments	0.95 (***)
num_comments x prior_interaction	0.97 (***)
prior_interaction x emotion	0.99
social_connection x emotion	0.92 (***)
main_team_member x emotion	0.93 (***)
AIC	**330951**

*$p < .05$. **$p < .01$. ***$p < .001$.

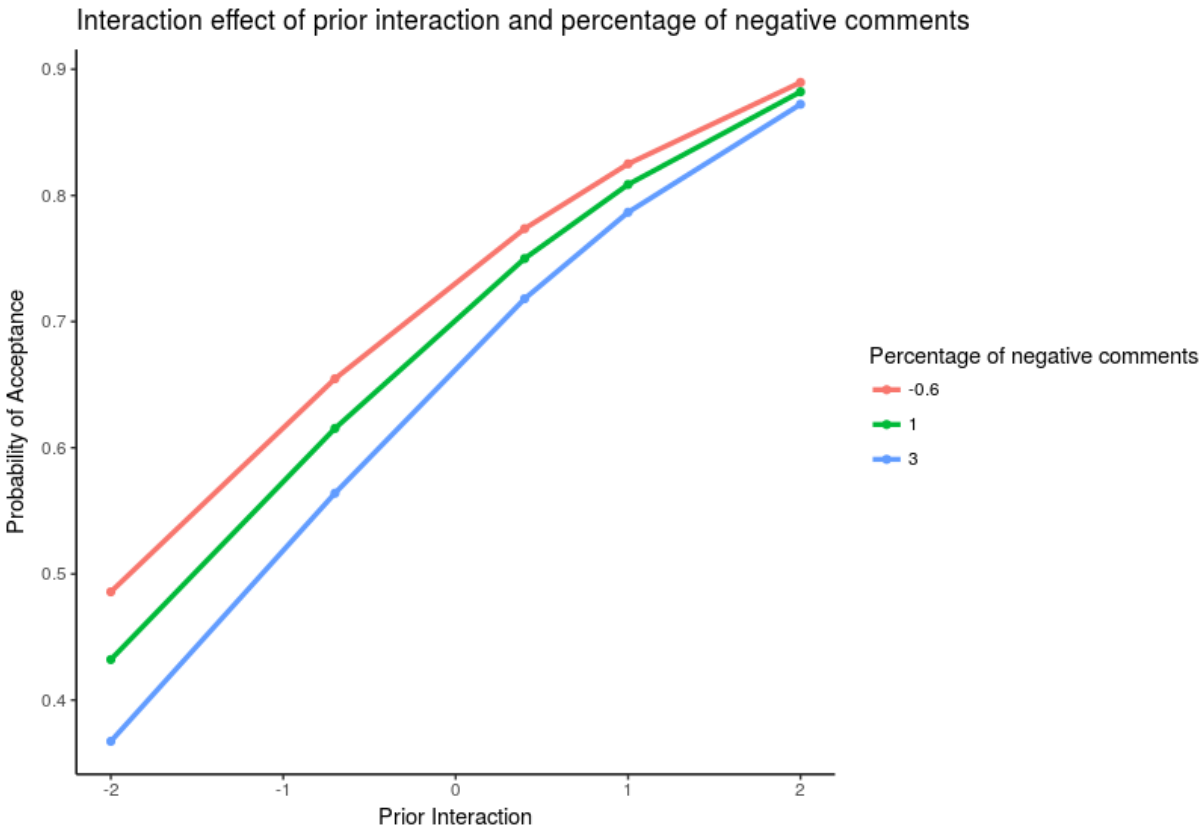
4.3 Discussion

In this section, we summarize the results, present the insights gained and implications of the result.

We found that the percentage of positive comments influences the pull request acceptance positively, and the percentage of negative comments influences the pull request acceptance negatively. The result verifies our initial hypothesis that emotions are present in the comments section. Hence, we can predict the pull request acceptance by knowing the emotional state of the pull request. This result is useful to the software engineering community as it provides a way to avoid undesirable emotional states. A developer who is more positive and prevents negative communication happening either from their side or from others' side helps contribute better. The result is supported by different research and surveys which show that negative emotions affect the general productivity of a developer [36, 91].

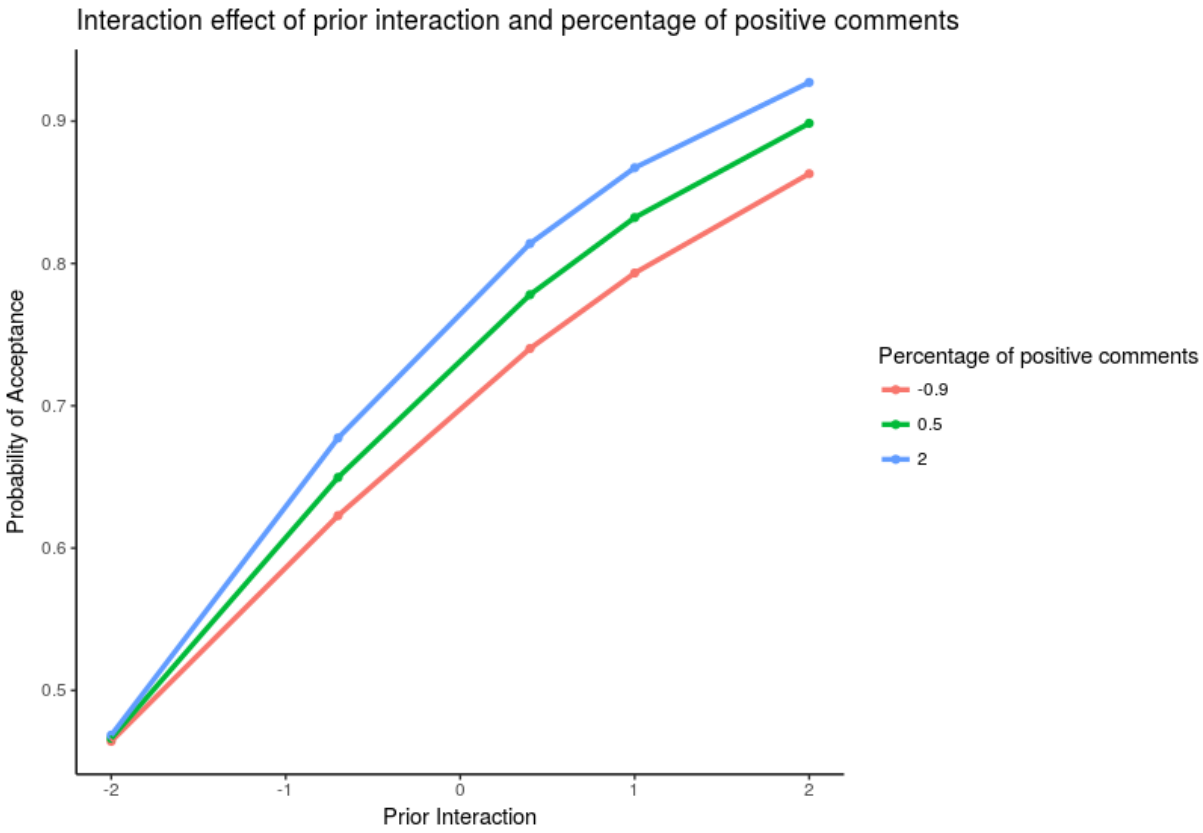
To understand more on role-specific emotions, we divided the research questions into additional parts that include requester's and closer's emotion. For RQ2, we only considered the percentage of positive and negative comments of the requester. The result was similar to RQ1; more positive comments indicated a higher pull request acceptance. We believe a more confident and an assuring requester works on the suggestions provided by the closer and other contributors to provide a working pull request. The result might also suggest strong coding skills of the requester due to which they answer more positively and confidently. A detailed analysis is needed to conclude the insights. Additionally, the experiment also measured the interaction effects of other social factors on emotion. We found significant interaction effects between social distance and percentage of requester's actual comments.

We also found significant effects between prior interaction and both the percentage of positive and negative comments, which indicates that moderating factors overpower the effects of emotion. Figure 4.3a suggests that as the interaction between the requester and the project increases, the effect of negative emotion is reduced (As seen by the distance between the different percentage of emotions reducing). With more prior interaction, developers might be gaining more trust, and hence, the project managers might not see having negative emotions negatively. A study by Park et al. [72] found that increase in group interaction increases the positive mood and reduces negative emotion among the group members. Hence it is also possible that the discussion thread is overall non-negative. We see an opposite effect for percentage of positive comments and prior interaction in figure 4.3b. As the requester indulges in more interaction in the project, the effect of positive



(a) Interaction plot of prior interaction and percentage of negative comments

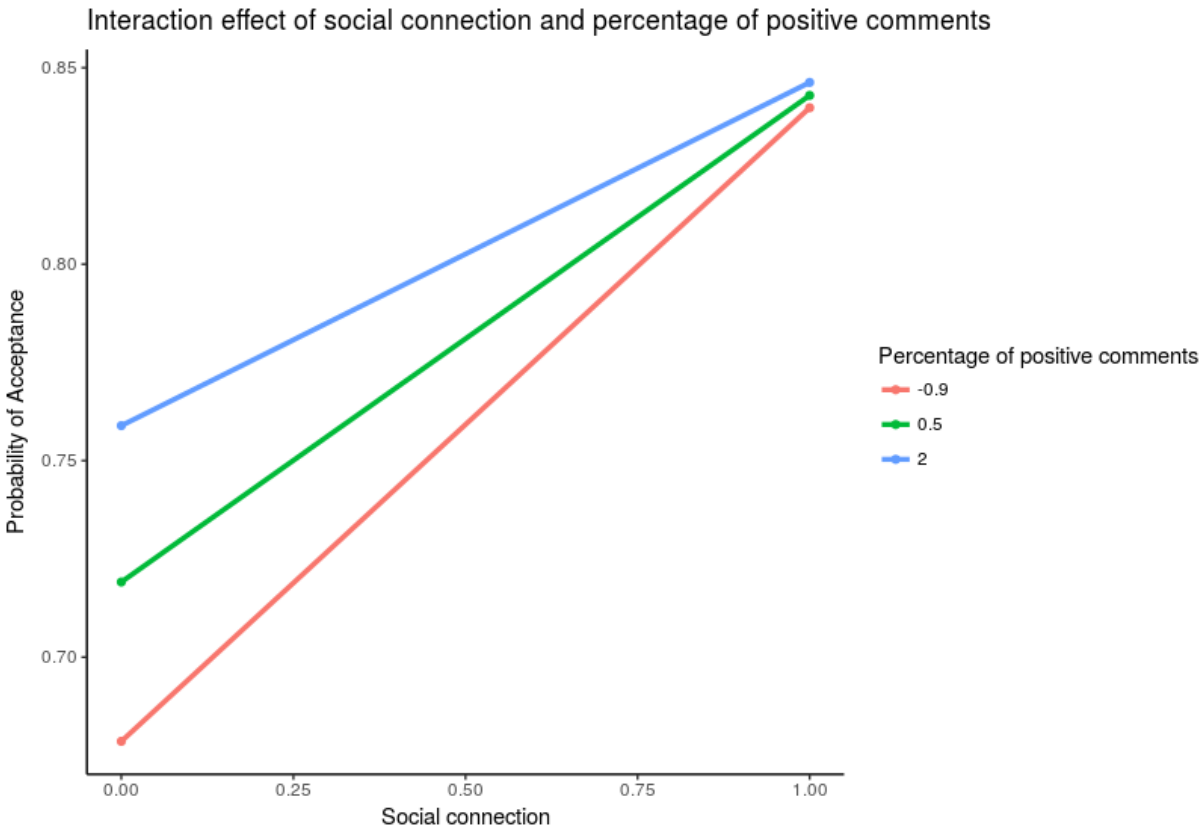
emotions increases. This effect can be because as requester becomes more trustworthy through interaction, the likelihood of pull request acceptance increases significantly. Hence, a positive comment from the requester might mean more than a negative comment. A higher prior interaction results in better pull request acceptance rate, which implies only those developers with better skills are indulged in the project. It is possible that their positive comment has higher importance than positive comments coming from developers who have fewer interactions. Finally, Figure 4.3c suggests that if the social distance is 1 the effect of positive emotion decreases, which indicates that social distance dominates emotional factors in importance. In RQ3, we wanted to understand how the emotion of requester's first response in the pull request discussion thread affects the pull request acceptance. The result suggests that a more positive first response from the developer leads to a higher likelihood of acceptance, which proves our hypothesis. We believe this happens when a requester responds more positively; it gives more assurance of a positive



(b) Interaction plot of prior interaction and percentage of positive comments

outcome of the discussion. For example, if a requester asks for clarification or suggests an improvement, a negative comment might question the requester's ability to correct the pull request. We did not find significant interaction effects of social factors.

In RQ-4 and RQ-5, we explore the closer's emotions and its effects in pull request acceptance. First, we modeled the percentage of positive and negative comments of the closer. Since closers are in charge of making the final decision, it is reasonable to believe that more negative emotion from the closer would be bad for pull request acceptance. The result support our hypothesis, and we found a higher percentage of positive comments is predictive of the pull request positively, and a higher percentage of negative comments is predictive of pull request acceptance negatively. Closers might tend to be more lenient and less negative if they know the requester personally or if the requester is socially respected. Hence we investigated the interaction effects of social factors. We found significant results for interaction effects between social distance and percentage of positive comments (Figure

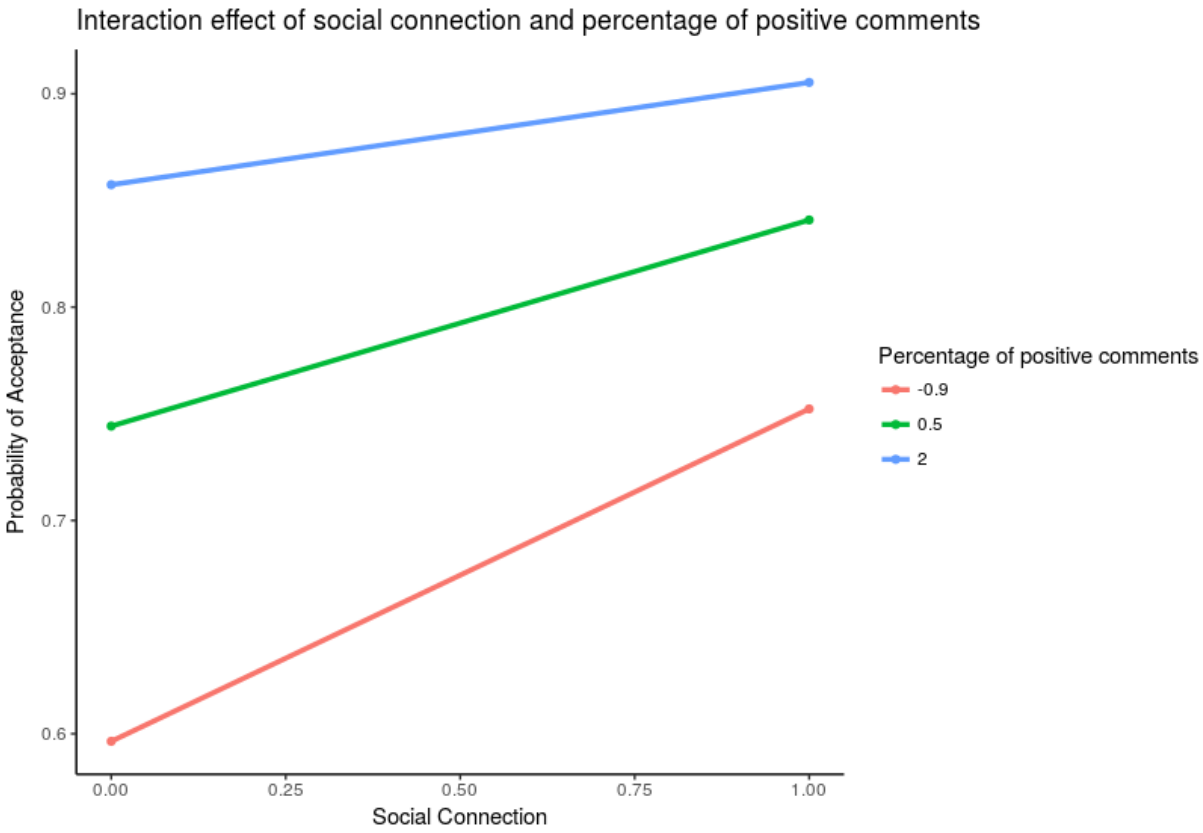


(c) Interaction plot of social distance and percentage of positive comments

Figure 4.3: Interaction plot for RQ-2

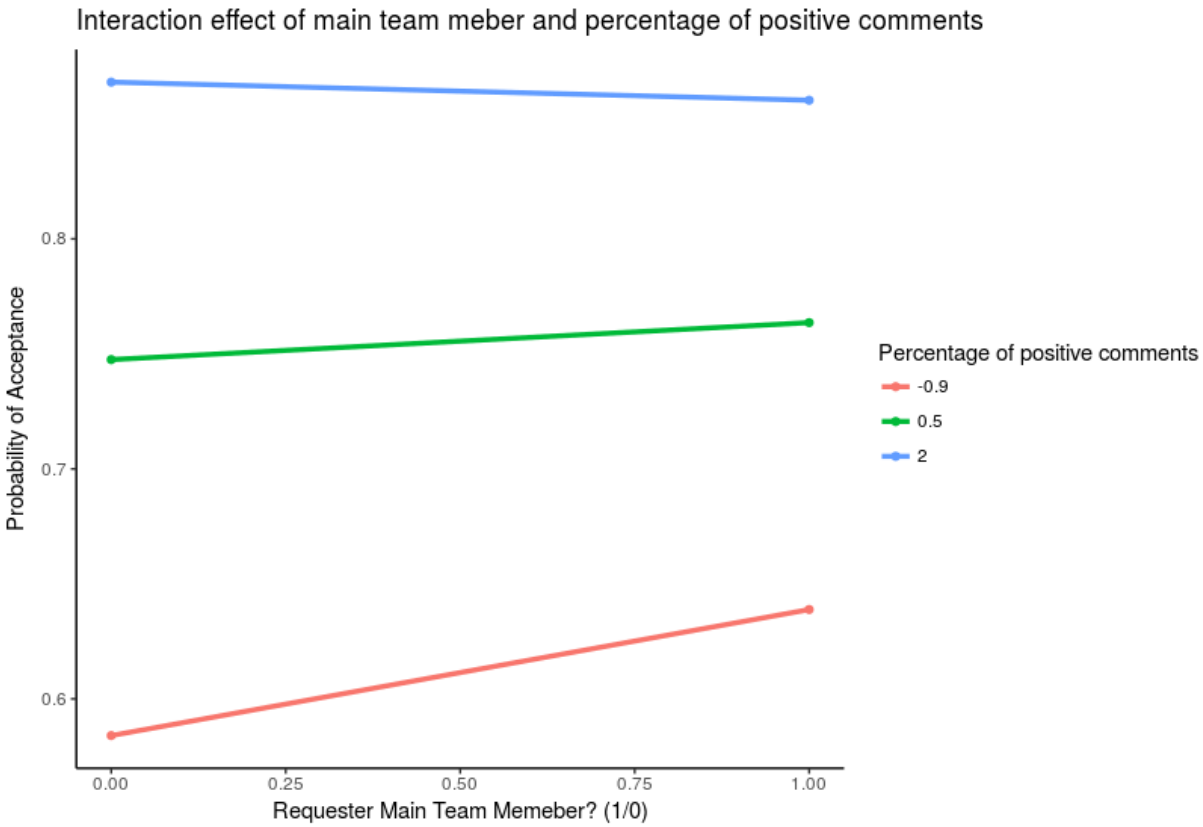
4.4a). We also found significant results for requester being the main team member and percentage of positive comments but similarly (Figure 4.4b). The effect of positive comments on pull request acceptance reduces when there is a positive social distance, and when the requester is the main team member. The effect might be due to closers emphasizing social respect more and hence not reacting negatively, which results in higher positive or neutral comments. This less variation of emotions provides lesser information to the model and hence reduces the effect of positive emotion.

In RQ 5, we wanted to know if the emotion of the first response from closer is predictive of pull request acceptance. We found that first comment from the closer is positively associated with the pull request acceptance. The result suggests that the closer might form a strong opinion about the pull request initially and do not change it that often. This



(a) Interaction plot of social connection and percentage of positive comments

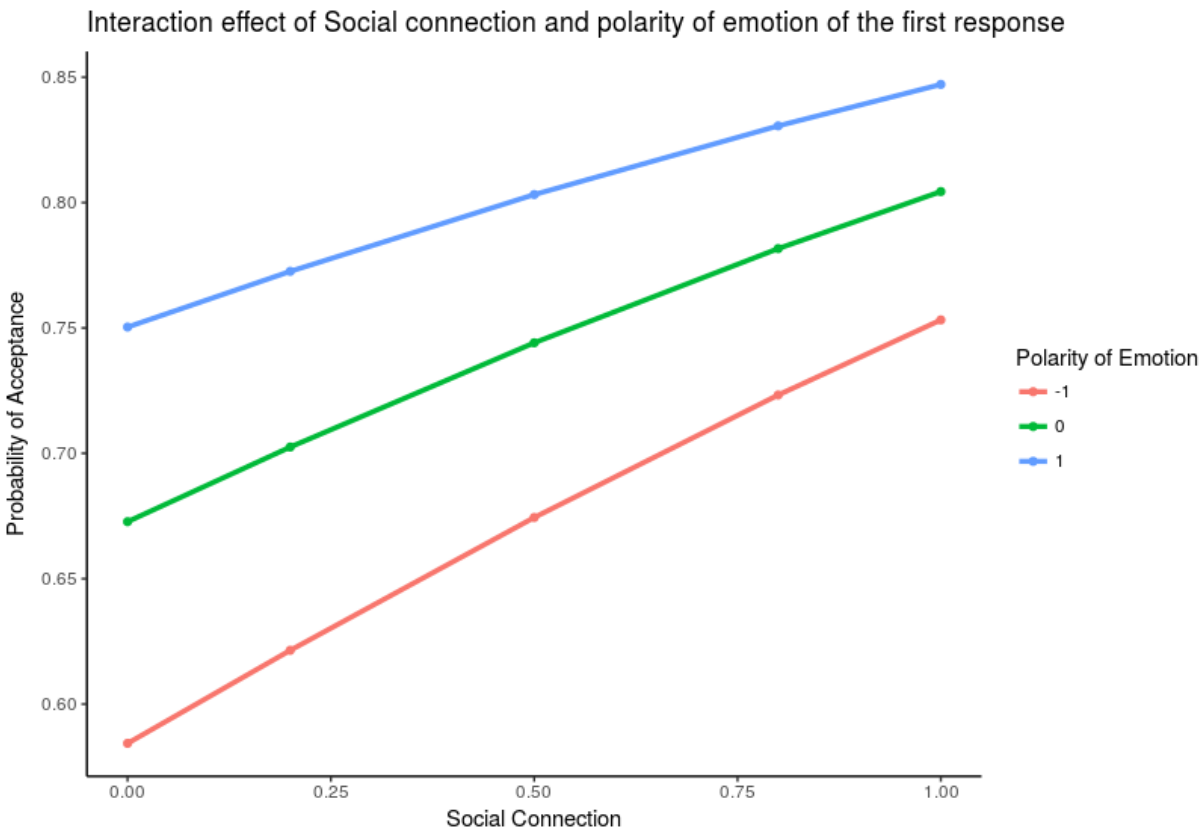
result is interesting for the community as it can help the developers make the pull request as good as possible and then submit for review. We also explored the interaction effects between the emotion of the first comment and other factors. The results showed there are significant effects between social distance and emotion and between requester being main team member and emotion. Figure 4.5a and 4.4b shows plots of significant interaction effects. This experiment hence proves that the overall emotion of the pull request can be used to determine the outcome of the pull request reliably along with other factors. We also found a positive requester has a better likelihood of acceptance. Additionally, we found that closers tend not to express negative emotions when the requester and closer have a social connection or when a requester is a main team member.



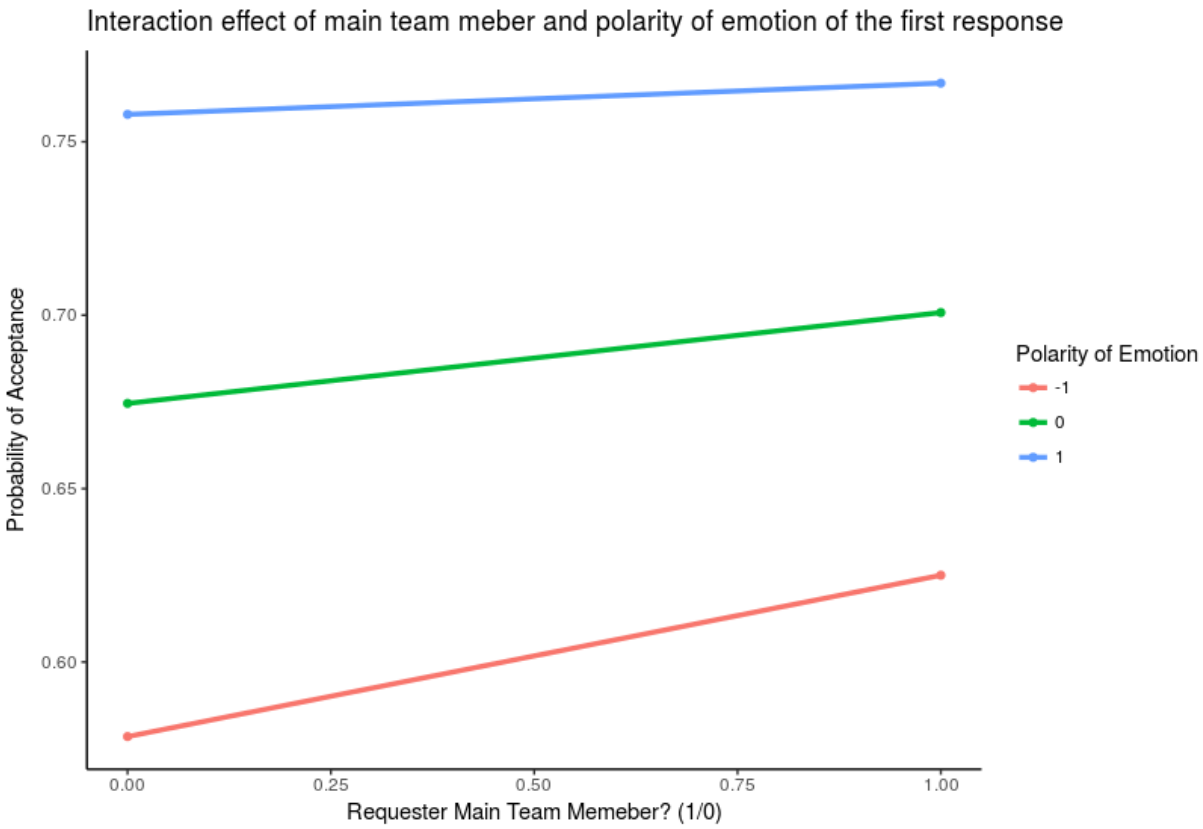
(b) Interaction plot of main team member and percentage of positive comments

Figure 4.4: Interaction plot for RQ-4

To summarise, in this chapter, we explore different emotional factors' predictive power. We look at the percentage of positive and negative comments of the requester, closer and the overall pull request. As hypothesised, we find a positive association of positive comments and negative association of negative comments with pull request acceptance. We also found social factors affecting emotional factors. Additionally, we found that the emotion associated with the first response of both closer and requester had information to predict pull request acceptance.



(a) Interaction plot of social connection and emotion polarity of the first response of closer



(b) Interaction plot of main team member and emotion polarity of the first response of closer

Figure 4.5: Interaction plot for RQ-5

Chapter 5

Conclusions

We presented empirical evidence showing the effects of developers' personality traits in pull request acceptance in GitHub. We first replicated Tsay et al.'s work [87], noting the importance of social and technical factors. Our results showed that the personality traits of developers significantly influence the likelihood of pull request acceptance. We also noted the absolute difference in personality traits between the requester and the closer results in positive effects, suggesting that diversity in personality is beneficial in open source projects. It is important to note that the personality traits had an effect size similar to other technical factors, but not to the extent of social factors. In sum, we observe requesters who are high on Openness, Conscientiousness, and low on Extraversion have a higher likelihood of getting the pull request accepted. Similarly, a closer who are high on Openness, Conscientiousness, Extraversion, and Neuroticism accepts more pull requests.

In the second experiment, we conducted an empirical analysis of the effects of developers' emotional response in pull request thread in pull request acceptance. Our results conclude that emotions play a significant role in predicting pull request acceptance. Throughout all the research questions, we found that a higher percentage of positive comments influences pull request positively and a higher percentage of negative comments in the pull request influences negatively. We also found that the emotion of the first response of both requester and closer has a positive association with the pull request acceptance. The results reinforce our belief that emotions are essential part of pull request discussions and contain enough information for predicting pull request acceptance. We also find the interaction effects of emotions of requester and closer to other social factors. Finally, we create a state of the art model for assigning emotion polarity to software engineering texts that outperforms previous methods on four different software engineering datasets.

Our work provides a stepping-stone for researchers to conduct further experiments, observing social and group dynamics in online collaborative environments. We believe that expensive qualitative studies, such as detailed developer interviews can now be conducted and can shed light on the underlying mechanisms for collaborative work. We conclude by recommending that the developer take time to understand the emotional state of the pull request and analyze them before responding. The findings of the thesis can be used to develop models that understand the emotional state and the behavior of the developers and help them communicate better.

References

[1] Archive stackexchange dump. https://archive.org/details/stackexchange.

[2] Bbc article. https://www.bbc.com/news/technology-45664640.

[3] Github api. https://developer.github.com/v3/. Accessed: 2018-08-30.

[4] Ibm personality insights service. https://console.bluemix.net/docs/services/personality-insights/science.html#science. Accessed: 2018-08-28.

[5] Octoverse 2018. https://octoverse.github.com/.

[6] Reporeaper. http://reporeapers.github.io/results/1.html. Accessed: 2018-08-28.

[7] Silvia T Acuña, Marta Gómez, and Natalia Juristo. How do personality, team processes and task characteristics relate to job satisfaction and software quality? *Information and Software Technology*, 51(3):627–639, 2009.

[8] Toufique Ahmed, Amiangshu Bosu, Anindya Iqbal, and Shahram Rahimi. Senticr: a customized sentiment analysis tool for code review interactions. In *Proceedings of the 32nd ieee/acm international conference on automated software engineering*, pages 106–111. IEEE Press, 2017.

[9] Pierre-Hadrien Arnoux, Anbang Xu, Neil Boyette, Jalal Mahmud, Rama Akkiraju, and Vibha Sinha. 25 tweets to know you: A new model to predict personality with social media. *arXiv preprint arXiv:1704.05513*, 2017.

[10] Nathan Baddoo and Tracy Hall. De-motivators for software process improvement: an analysis of practitioners views. *Journal of Systems and Software*, 66(1):23–33, 2003.

[11] Sebastian Baltes, Lorik Dumani, Christoph Treude, and Stephan Diehl. Sotorrent: Reconstructing and analyzing the evolution of stack overflow posts. In *Proceedings of the 15th International Conference on Mining Software Repositories*, pages 319–330. ACM, 2018.

[12] Murray R Barrick and Michael K Mount. The big five personality dimensions and job performance: a meta-analysis. *Personnel psychology*, 44(1):1–26, 1991.

[13] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

[14] Blerina Bazelli, Abram Hindle, and Eleni Stroulia. On the personality traits of stackoverflow users. In *Software maintenance (ICSM), 2013 29th IEEE international conference on*, pages 460–463. IEEE, 2013.

[15] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[16] Fabio Calefato, Giuseppe Iaffaldano, Filippo Lanubile, and Bogdan Vasilescu. On developers' personality in large-scale distributed projects: The case of the apache ecosystem. In *Proceedings of the 13th International Conference on Global Software Engineering*, ICGSE '18, pages 92–101, New York, NY, USA, 2018. ACM.

[17] Fabio Calefato and Filippo Lanubile. Affective trust as a predictor of successful collaboration in distributed software projects. In *2016 IEEE/ACM 1st International Workshop on Emotional Awareness in Software Engineering (SEmotion)*, pages 3–5. IEEE, 2016.

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

[19] Fabio Calefato, Filippo Lanubile, and Nicole Novielli. Emotxt: a toolkit for emotion recognition from text. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 79–80. IEEE, 2017.

[20] Luiz Fernando Capretz and Faheem Ahmed. Making sense of software development and personality types. *IT professional*, 12(1), 2010.

[21] Raymond B Cattell. Personality: A systematic theoretical and factual study. 1950.

[22] Jilin Chen, Yuqing Ren, and John Riedl. The effects of diversity on group productivity and member withdrawal in online volunteer groups. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 821–830. ACM, 2010.

[23] Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968.

[24] Paul T Costa and Robert R McCrae. The neo personality inventory. 1985.

[25] Paul T Costa Jr, Robert R McCrae, and David A Dye. Facet scales for agreeableness and conscientiousness: A revision of the neo personality inventory. *Personality and individual Differences*, 12(9):887–898, 1991.

[26] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. Social coding in github: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*, pages 1277–1286. ACM, 2012.

[27] Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A computational approach to politeness with application to social factors. *arXiv preprint arXiv:1306.6078*, 2013.

[28] Giuseppe Destefanis, Marco Ortu, David Bowes, Michele Marchesi, and Roberto Tonelli. On measuring affects of github issues commenters. 2018.

[29] Giuseppe Destefanis, Marco Ortu, Steve Counsell, Stephen Swift, Michele Marchesi, and Roberto Tonelli. Software development: do good manners matter? *PeerJ Computer Science*, 2:e73, 2016.

[30] Prasun Dewan. Towards emotion-based collaborative software engineering. In *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 109–112. IEEE, 2015.

[31] Colin G DeYoung, Lena C Quilty, and Jordan B Peterson. Between facets and domains: 10 aspects of the big five. *Journal of personality and social psychology*, 93(5):880, 2007.

[32] John M Digman. Personality structure: Emergence of the five-factor model. *Annual review of psychology*, 41(1):417–440, 1990.

[33] Jin Ding, Hailong Sun, Xu Wang, and Xudong Liu. Entity-level sentiment analysis of issue comments. In *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*, pages 7–13. ACM, 2018.

[34] Georgios Gousios. The ghtorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, pages 233–236, Piscataway, NJ, USA, 2013. IEEE Press.

[35] Georgios Gousios, Martin Pinzger, and Arie van Deursen. An exploratory study of the pull-based software development model. In *Proceedings of the 36th International Conference on Software Engineering*, pages 345–355. ACM, 2014.

[36] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. Software developers, moods, emotions, and performance. *arXiv preprint arXiv:1405.4422*, 2014.

[37] Emitza Guzman, David Azócar, and Yang Li. Sentiment analysis of commit comments in github: an empirical study. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 352–355. ACM, 2014.

[38] Emitza Guzman and Bernd Bruegge. Towards emotional awareness in software development teams. In *Proceedings of the 2013 9th joint meeting on foundations of software engineering*, pages 671–674. ACM, 2013.

[39] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[40] Robert Hogan. What is personality psychology? *Psychological Inquiry*, 9(2):152–153, 1998.

[41] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 2017.

[42] Jeremy Howard et al. fastai. https://github.com/fastai/fastai, 2018.

[43] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[44] Md Rakibul Islam and Minhaz F Zibran. Towards understanding and exploiting developers' emotional variations in software engineering. In *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 185–192. IEEE, 2016.

[45] Md Rakibul Islam and Minhaz F Zibran. Leveraging automated sentiment analysis in software engineering. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, pages 203–214. IEEE, 2017.

[46] Robbert Jongeling, Proshanta Sarkar, Subhajit Datta, and Alexander Serebrenik. On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering*, 22(5):2543–2584, 2017.

[47] Romualdo Alves Pereira Junior and Diana Inkpen. Using cognitive computing to get insights on personality traits from twitter messages. In *Canadian Conference on Artificial Intelligence*, pages 278–283. Springer, 2017.

[48] John Karn and Tony Cowling. A follow up study of the effect of personality on the performance of software engineering teams. In *Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering*, pages 232–241. ACM, 2006.

[49] John S Karn, Sharifah Syed-Abdullah, Anthony J Cowling, and Mike Holcombe. A study into the effects of personality type and methodology on cohesion in software engineering teams. *Behaviour & Information Technology*, 26(2):99–111, 2007.

[50] Sherlock A Licorish and Stephen G MacDonell. Personality profiles of global software developers. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, page 45. ACM, 2014.

[51] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, Michele Lanza, and Rocco Oliveto. Sentiment analysis for software engineering: How far can we go? In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, pages 94–104. IEEE, 2018.

[52] Elizabeth Mannix and Margaret A Neale. What differences make a difference? the promise and reality of diverse teams in organizations. *Psychological science in the public interest*, 6(2):31–55, 2005.

[53] Leandro Soriano Marcolino, Albert Xin Jiang, and Milind Tambe. Multi-agent team formation: diversity beats strength? In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 279–285. AAAI Press, 2013.

[54] Jennifer Marlow, Laura Dabbish, and Jim Herbsleb. Impression formation in online peer production: activity traces and personal profiles in github. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 117–128. ACM, 2013.

[55] Allen Marshall, Rose F Gamble, and Matthew L Hale. Outcomes of emotional content from agile team forum posts. In *Proceedings of the 1st International Workshop on Emotion Awareness in Software Engineering*, pages 6–11. ACM, 2016.

[56] Luis G Martínez, Antonio Rodríguez-Díaz, Guillermo Licea, and Juan R Castro. Big five patterns for software engineering roles using an anfis learning approach with ram-set. In *Mexican International Conference on Artificial Intelligence*, pages 428–439. Springer, 2010.

[57] Robert R McCrae and Paul T Costa. Updating norman's" adequacy taxonomy": Intelligence and personality dimensions in natural language and in questionnaires. *Journal of personality and social psychology*, 49(3):710, 1985.

[58] Robert R McCrae and Paul T Costa. Validation of the five-factor model of personality across instruments and observers. *Journal of personality and social psychology*, 52(1):81, 1987.

[59] Robert R McCrae and Paul T Costa Jr. Reinterpreting the myers-briggs type indicator from the perspective of the five-factor model of personality. *Journal of personality*, 57(1):17–40, 1989.

[60] RR McCrae. Creativity, divergent thinking, and openness to experience. *Journal of personality and social psychology*, 52(6):1258–1265, 1987.

[61] RR McCrae, PT Costa, GJ Boyle, G Matthews, and DH Saklofske. *Sage handbook of personality theory and assessment*. Boyle, 2008.

[62] Albert Mehrabian. Analysis of the big-five personality factors in terms of the pad temperament model. *Australian journal of Psychology*, 48(2):86–92, 1996.

[63] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*, 2017.

[64] Nuthan Munaiah, Steven Kroh, Craig Cabrey, and Meiyappan Nagappan. Curating github for engineered software projects. *Empirical Software Engineering*, 22(6):3219–3253, 2017.

[65] Alessandro Murgia, Parastou Tourani, Bram Adams, and Marco Ortu. Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In *Proceedings of the 11th working conference on mining software repositories*, pages 262–271. ACM, 2014.

[66] Isabel Briggs Myers, Mary H McCaulley, and Robert Most. *Manual: A guide to the development and use of the Myers-Briggs Type Indicator*, volume 1985. Consulting Psychologists Press Palo Alto, CA, 1985.

[67] Weiting Ng and Ed Diener. Personality differences in emotions: Does emotion regulation play a role? *Journal of Individual Differences*, 30(2):100–106, 2009.

[68] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. Towards discovering the role of emotions in stack overflow. In *Proceedings of the 6th international workshop on social software engineering*, pages 33–36. ACM, 2014.

[69] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. The challenges of sentiment detection in the social programmer ecosystem. In *Proceedings of the 7th International Workshop on Social Software Engineering*, pages 33–40. ACM, 2015.

[70] Nicole Novielli, Daniela Girardi, and Filippo Lanubile. A benchmark study on sentiment analysis for software engineering research. In *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*, pages 364–375. IEEE, 2018.

[71] Marco Ortu, Alessandro Murgia, Giuseppe Destefanis, Parastou Tourani, Roberto Tonelli, Michele Marchesi, and Bram Adams. The emotional side of software developers in jira. In *Proceedings of the 13th International Conference on Mining Software Repositories*, pages 480–483. ACM, 2016.

[72] Ernest S Park and Verlin B Hinsz. Group interaction sustains positive moods and diminishes negative moods. *Group Dynamics: Theory, Research, and Practice*, 19(4):290, 2015.

[73] Oscar Hernán Paruma-Pabón, Fabio A González, Jairo Aponte, Jorge E Camargo, and Felipe Restrepo-Calle. Finding relationships between socio-technical aspects and personality traits by mining developer e-mails. In *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 8–14. ACM, 2016.

[74] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[75] David J Pittenger. Cautionary comments regarding the myers-briggs type indicator. *Consulting Psychology Journal: Practice and Research*, 57(3):210, 2005.

[76] Daniel Pletea, Bogdan Vasilescu, and Alexander Serebrenik. Security and emotion: sentiment analysis of security discussions on github. In *Proceedings of the 11th working conference on mining software repositories*, pages 348–351. ACM, 2014.

[77] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[78] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

[79] Achyudh Ram and Meiyappan Nagappan. Supervised sentiment classification with cnns for diverse se datasets. *arXiv preprint arXiv:1812.09653*, 2018.

[80] Ayushi Rastogi and Nachiappan Nagappan. On the personality traits of github contributors. In *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, pages 77–86. IEEE, 2016.

[81] Peter C Rigby and Ahmed E Hassan. What can oss mailing lists tell us? a preliminary psychometric text analysis of the apache developer mailing list. In *Proceedings of the fourth international workshop on mining software repositories*, page 23. IEEE Computer Society, 2007.

[82] Vinayak Sinha, Alina Lazar, and Bonita Sharif. Analyzing developer sentiment in commit logs. In *Proceedings of the 13th International Conference on Mining Software Repositories*, pages 520–523. ACM, 2016.

[83] Daricélio Moreira Soares, Manoel Limeira de Lima Júnior, Leonardo Murta, and Alexandre Plastino. Acceptance factors of pull requests in open-source projects. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 1541–1546. ACM, 2015.

[84] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[85] Chakkrit Tantithamthavorn and Ahmed E Hassan. An experience report on defect modelling in practice: Pitfalls and challenges. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, pages 286–295. ACM, 2018.

[86] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012.

[87] Jason Tsay, Laura Dabbish, and James Herbsleb. Influence of social and technical factors for evaluating contribution in github. In *Proceedings of the 36th international conference on Software engineering*, pages 356–366. ACM, 2014.

[88] Daniel Varona, Luiz Fernando Capretz, Yadenis Piñero, and Arif Raza. Evolution of software engineers' personality profile. *ACM SIGSOFT Software Engineering Notes*, 37(1):1–5, 2012.

[89] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066, 2013.

[90] Yi Wang. Building the linkage between project managers' personality and success of software projects. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 410–413. IEEE, 2009.

[91] Michal R Wrobel. Emotions in the software development process. In *2013 6th International Conference on Human System Interactions (HSI)*, pages 518–523. IEEE, 2013.

[92] Yue Yu, Huaimin Wang, Vladimir Filkov, Premkumar Devanbu, and Bogdan Vasilescu. Wait for it: Determinants of pull request evaluation latency on github. In *Mining software repositories (MSR), 2015 IEEE/ACM 12th working conference on*, pages 367–371. IEEE, 2015.