On the Relation of Sentiment with Difficulty and Popularity in Stackoverflow

by

SRILAKSHMI SRIPATHI

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

2020

Oakland University
Rochester, Michigan

Thesis Advisory Committee:

Dr. Mehdi Bagherzadeh, Ph.D., Chair
Dr. Debatosh Debnath, Ph.D.
Dr. Hua Ming, Ph.D.

Dedicated to my friends and family

ACKNOWLEDGMENTS

I want to thank everyone who helped me with various aspects of conducting this research. My special thanks go to my advisor, Dr. Mehdi Bagherzadeh, for helping me to pursue the research and for his guidance, patience, and support throughout this research. His insights and words of encouragement will always resonate in my life. I thank the Department of Computer Science and Engineering and the Research Office at Oakland University for supporting and recognizing our research. I wish to acknowledge the support and unyielding love of my family, my husband, Hari Kishan, my mother, Padmavathi, my in-laws, Rajendra Prasad, and Indira Krishna. I would also like to thank my classmate Andrew Walwema and my roommate Thanaa Ali Al-Khawaja for being a helpful research partner. They all kept me going on, and this work would not have been possible without their encouragement.

Srilakshmi Sripathi

ABSTRACT

On the Relation of Sentiment with Difficulty and Popularity in Stackoverflow

by

Srilakshmi Sripathi

Adviser: Dr. Mehdi Bagherzadeh, Ph.D.

In the current software development environment, a software developer's sentiment towards a challenging task may have an impact on their productivity and the overall quality of their software. Understanding the role of the sentiment can help developers and research communities improve developer's productivity. In this thesis, we conducted a large-scale study by focusing our attention on software developer's sentiment in big data and concurrency software development. Firstly, we use the previous works topic categorization containing 150K questions and answers from 29 big data topics and 250K questions and answers from 27 concurrency topics. Furthermore, we utilize the five measures used in calculating popularity and difficulty metrics by those previous works to understand the role of sentiment. Secondly, we perform sentiment analysis by using the tool Senti4SD to calculate sentiment polarity for each question and answer from those previous works. Finally, using Kendall bivariant correlation, we calculate correlations between popularity with the sentiment and difficulty with the sentiment. Our results indicate that there is a strong inverse correlation between the popularity metric's average favorites measure with the negative sentiment in both concurrency and big data datasets. Considering concurrency findings, we found a negative correlation exists between the popularity metric's average favorites measure with the negative sentiment. Furthermore, there exists a positive correlation between the popularity metric's average scores measure with the neutral sentiment. While our big data

findings, imply an inverse and a direct correlation between negative and neutral sentiment with the popularity metric's average favorites measure. Additionally, there exists a direct and an inverse correlation between the negative and the neutral sentiment with the difficulty metric's percentage of questions with accepted answers measure.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

"Happy software developers solve problems better" - Graziotin et al. [3]

In the current software development environment, a software developer's sentiment towards a task may have an impact on their productivity and the overall quality of the software they develop. Understanding this role of the sentiment can help developers and research communities to help in the improvement of the developer's overall throughput.

For the last three decades, improving software quality and software developers' productivity has been a focus; amongst different factors, the people factor has been considered one of the vital factors [3, 4]. So, in recent years, the software engineering community has been focused on software developer's sentiment, as human emotions are volatile and can have a direct impact on their productivity. For instance, developers with a positive sentiment are more likely to be creative and have better analytical problem-solving skills, as illustrated by Graziotin et al. [3]. While software developers with positive sentiment can be more productive, people with negative sentiment are likely to depart from the project sooner [5].

Currently, the Stackoverflow [6] website has 11 million software developers as its participants. The platform has 46 million questions and answers posted by those participants, with an average of 6.8 thousand new questions posted daily, and with 10 million visitors to Stackoverflow a day. To enhance their knowledge, developers ranging from beginners to experts often visit the Stackoverflow website to post questions and receive answers from the subject matter experts in the field. For this reason, Stackoverflow is an excellent resource available to study software developers and the problems they encounter. Stackoverflow users often ask questions about software topics like Software Development, Programming Languages, and more. If a user posts a question that does not associate with Software topics, the questions get redirected by the Stackoverflow platform to the appropriate community within StackExchange. For instance, if a user posts a question related to the educational topic to the Stackoverflow platform, the question gets redirected to the Academia community, which is a subcommunity of StackExchange.

To understand the role of the software developer's sentiment, we perform sentiment analysis, which is a study of lexicons within a text to determine the subjectivity and polarity. Subjectivity helps determine if the text is emotionally loaded or if it is neutrally toned. On the other hand, polarity checks if the meaning of the text is positive, negative, or neutral. The sentiment for the entire text is determined by analyzing the values for subjectivity and polarity for each lexicon. We are interested in analyzing the role of the sentiment with software topics that are difficult to understand and concepts that are currently popular.

In this thesis, we conducted a large-scale study on the Stackoverflow questions and answers for big data and concurrency software development. We focused our attention on the software developer's sentiment and its correlations with popularity and difficulty by answering the research questions ***R1*** to ***R4***, as shown below. To perform our analysis, we rely on two previous works on concurrency and big data [1, 2]. We considered concurrency topics since developers find it challenging to develop concurrency programs, while big data topics are currently popular concepts. The concurrency dataset [1] has 250K questions and answers with 29 concurrency topics, and the study calculates the difficulty and popularity percentages. Similarly, in the big data dataset, the dataset has 150K questions and answers with 27 big data topics, and the study calculates the difficulty and popularity percentages.

We study the relationship between sentiment with both popularity and difficulty metrics by computing the sentiment for each topic in the two datasets and find if a correlation exists to understand the following questions:

R1. Is there a relation between sentiment with popularity and sentiment with difficulty in concurrency topics?

R2. Is there a relation between sentiment with popularity and sentiment with difficulty in big data topics?

R3. What sentiment do concurrency and big data developers express towards a common topic?

R4. Is there a commonality in the correlations found in concurrency and big data datasets?

This thesis's chapters are structured as follows. In Chapter 2, we discuss the dataset, sentiment analysis tool Senti4SD, and finally, we mention the steps we used to perform the analysis. In Chapter 3, we present the results of the sentiment analysis for the concurrency and big data datasets and examine the correlations. In Chapter 4, we present the implications of this study. In Chapter 5, we review works that are related to this study. Finally, we present the conclusion and the future work in Chapter 6.

CHAPTER TWO

METHODOLOGY

In this chapter, we examine the methods used in our analysis in detail to answer the research questions ***R1*** to ***R4***. We begin with the background of the Stackoverflow questions and answers. Next, our methodology to perform sentiment analysis. Finally, we discuss the methods to find whether correlations exist within the datasets.

2.1 Stackoverflow Questions and Answers

It is essential to understand the structure of Stackoverflow questions and answers and their usage in our analysis. When a registered Stackoverflow participant posts a question related to software topics, the post generates question metadata. Similarly, when a participant answers the posted question, it generates answer metadata. When the participant requesting the question receives a relevant answer, the most relevant answer becomes an accepted answer. The Stackoverflow participant and platform are the two sources that generate the question metadata. The participant generates the question title, descriptive question body, relevant tags, and accepted answer. While the Stackoverflow platform generates the unique identifier, date and time of the post, the number of favorites, score, and the number of views to the question.

Similarly, the answer metadata has two sources, where the participant generates the answer body, and the platform generates the answer identifier and date. When participants post duplicate questions, the Stackoverflow platform often merges the two questions into one. If a participant considers and marks an answer as a relevant solution for this problem, then the Stackoverflow platform considers that answer as an accepted answer. For an example question and answer metadata, refer to the Table 2.1 and the Table 2.2 and note the metadata updates with time.

Table 2.1: Stackoverflow Sample Question

Question Metadata	**Description**	**Example**
question id	unique identifier	47613802
title	title given by user	how to create threads dynamically?
body	question content	I want to create certain number of threads in my program where the number of threads to be created is provided by the user at run-time
date	time and date of post	2017-12-02 23:44:13.160
tags	Stackoverflow tags	multithreading, java, dynamic
favorites	count of user likes	79
score	like and dislike score	150
views	total views	454
accepted answer	user accepted answer	47613819

Table 2.2: Stackoverflow Sample Answer

Answer Metadata	**Description**	**Example**
answer id	a unique identifier	47613819
date	time and date of post	2017-12-02 23:47:28.157
body	an answer to question	there are a number of ways to do this. A for loop is the easiest your runnable couple be something like this

2.2 Concurrency and Big Data Datasets

To perform our sentiment analysis study, we begin by examining the data for concurrency [1] and big data [2] by using the two datasets from the previous publications. These two papers analyze the Stackoverflow question and answer posts from the Stackoverflow official data dump, which is publicly available at [6]. In this section, we discuss the contents of the two datasets, as shown in Figure 2.2; these are the questions and answers, the topics, and the measures for popularity and difficulty metrics.

2.2.1 Concurrency and Big Data Questions and Answers

For concurrency dataset, we utilized Ahmed and Bagherzadeh's work [1], which analyzes the Stackoverflow data over 9 years from August 2008 until December 2017. This dataset includes 38,485,046 questions and answers posted by 3,589,412 software developer participants of the Stackoverflow platform. Among these posts, 14,995,834 (39%) are questions and 23,489,212 (61%) are answers. From 38,485,046 questions and answers, the study concluded by identifying 245,576 (64%) of questions and answers belonging to the topic of concurrency. In those concurrency posts of the total of 245,576 questions and answers, 156,812 (64%) belong to question posts and 88,764 (36%) belonging to answer posts.

To study big data developers, we utilized the data from Bagherzadeh and Khatchadourian [2] work, which categorizes Stackoverflow questions and answers as big data topics. The input dataset for the study includes 42,850,541 Stackoverflow questions and answers posted by 4,142,516 software developers over a 10-year duration from August 2008 to December 2018. The study identifying 157,525 questions and answers belonging to big data topics of those, 112,948 (72%) are questions, and 44,577 (28%) are answers. Our study aggregates both papers, and our dataset input has 403,413 questions and answers containing 269,818 (67%) questions and 133,595 (33%) answers. To see the statistics of our dataset, refer to Table 2.3. We perform sentiment analysis on 403,413 questions and answers to answer our research questions.

2.2.2 Concurrency and Big Data Topics

To study the sentiment of concurrency and big data topics, we utilize the topic categorization conducted by the papers [1, 2]. These two works categorize the questions and answers using Latent Dirichlet Allocation (LDA) [7] and an open card sort. Topic modeling using LDA means assigning related topic names over a normal distribution based on keywords existing in the question and answer. And, an open card sort means that topic names are not predefined but assigned in the topic labeling process. The topic labeling process the studies [1, 2] follow, is that they manually sample through 15 random posts from individual topics. By iteratively refining associated topics, 27 topics names were manually assigned to 245,576 concurrency questions and answers. To see the concurrency topic names and their descriptions, refer to Table 2.4. For 157,525 big data questions and answers, the study [2] assigns them to 28 big data topics. For detailed big data topic names and descriptions, refer to Table 2.5. We perform sentiment analysis to assess sentiments for each topic in concurrency and big data datasets.

Table 2.3: Statistics of Concurrency and Big Data Dataset

Dataset	**No. of Q&As**	**No. of Questions**	**No. of Answers**
concurrency	245,576	156,812 (64%)	88,764 (36%)
big data	157,525	112,948 (72%)	44,577 (28%)
Total	**403,413**	**269,818 (67%)**	**133,595 (33%)**

Table 2.4: Concurrency Topics From the Previous Work [1]

No.	Topic	Q & As Topic Words
1	basic concepts	code understand read find edit make
2	task parallelism	task schedul async asynchron parallel
3	producer-consumer concurrency	queue consum produc process buffer block
4	parallel computing	parallel node loop comput openmp mpi
5	process life cycle management	process child parent termin fork kill start
6	python multiprocessing	python script run multiprocess parallel php
7	thread life cycle management	thread execut background separ join finish
8	thread sharing	variabl pass call pointer pthread global
9	thread scheduling	loop time wait stop run start sleep set check
10	thread pool	pool job process task limit threadpool
11	concurrent collections	list arrai map element collect iter kei add
12	thread safety	thread java safe multipl multi multithread
13	locking	lock mutex wait condit semaphor acquir
14	memory consistency	read write variabl atom cach share synchron
15	entity management	concurr session spring entiti transact actor
16	database management systems	databas tabl queri row record lock insert sql
17	file management	file read write log line open stream directori
18	object-oriented concurrency	object class method creat static access
19	web concurrency	servic web server applic user app net http
20	event-based concurrency	event signal handler timer callback call slot
21	mobile concurrency	android app view updat frame asynctask
22	client-server concurrency	connect send socket receiv port read
23	data scraping	data time load download problem
24	runtime speedup	cpu core time performance
25	unexpected output	code work run output problem print line
26	irreproducible behavior	error run problem applic compil crash
27	GUI	updating gui windows applications

Table 2.5: Big Data Topics From the Previous Work [2]

No.	Topic	Q & As Topic Words
1	memory management	memori executor task job spark run core
2	PIG	pig scripts oozi job workflow action run error
3	connection	connect hadoop server instal cloudera
4	dataset load&store	file data spark read json csv parquet load
5	data organization	partit join data number case oper kei set
6	scala spark	scala spark org apach java sql appli anonfun
7	pyspark	python pyspark spark zeppelin error run
8	dataframe	column datafram row spark pyspark data
9	file distribution	node hadoop cluster namenod hdf datanod
10	general programming	class function method object call code type
11	debugging	error code run work problem issu except fail
12	hbase	hbase tabl row region zookeep client scan
13	job management	spark run cluster job applic submit yarn
14	file format	file line read input split text block size data
15	file management	file hdf directori path hadoop folder local
16	string	string type field arrai null column valu
17	database import&export	sqoop cassandra jdbc mysql databas connector
18	stream processing	stream spark kafka data messag batch process
19	basic concepts	data hadoop process system databas solut
20	date&time	date time timestamp data month event record
21	logging	info flume log java org channel apach sink
22	performance	data perform memori process size run million
23	text search	document mongodb queri collect elastic search
24	dependency management	jar hadoop version spark java error run depend
25	debugging	java org apach hadoop lang run mapr util
26	machine learning	model spark vector mahout train featur data
27	hive analysis	hive tabl queri creat data sql select insert
28	mapreduce	kei group count reduc hadoop map reduc
29	RDD	rdd spark transform code function scala map

2.2.3 Concurrency and Big Data Popularity and Difficulty

We utilized popularity and difficulty metrics for the concurrency and big data topics from [1, 2] to determine the correlation between sentiment with popularity and sentiment with difficulty. The authors evaluated popularity and difficulty metrics using questions and answers metadata of the Stackoverflow, given in Table 2.1 and the Table 2.2. The popularity metric was calculated by 3 well-known measures, which are the average views, average favorites, and average scores. Likewise, they evaluated the difficulty metric using 2 well-known measures, which are the percentage of questions without accepted answers and the hours to receive an accepted answer.

To be more specific, the definitions of the popularity metrics are as follows; they defined average views measure [8, 9, 10, 11] as the average number of visits to the posted question to the number of questions in a topic. They considered the average views measure because Stackoverflow allows both registered and unregistered users to visit questions. We consider average views because the number of views to the Stackoverflow platform by the unregistered users is higher in number compared to registered users [12]. Secondly, as defined in [8, 9, 13, 11], the average favorites measure uses the question metadata's favorites counter by taking the average of it within a topic. To be more specific, the average favorites is the total number of registered users who mark the question as to their favorites. Finally, as defined in [8, 9, 13, 11], the average scores measure is calculating the average of the question metadata's scores counter within a topic. To be more specific, the average scores is the total number of registered users who like or dislike the question. For more details about popularity metrics, refer to Table 2.6.

On the other hand, the difficulty metric defined 2 measures that are as follows. The first measure the percentage of questions that do not have accepted answers within a topic [10, 14, 11]. The second measure checks the average time that is taken for a question in

Table 2.6: Popularity and Difficulty Metrics

Metrics	Measures	Description
popularity	average views	an average number of views for all the questions in a topic the questions in a topic
	average favorites	total average of users marking the question as their as their favorite in a topic
	average scores	average of raw score for all the questions within a topic
difficulty	%w/o acc. answer	percentage of questions without accepted answers
	hrs to acc. answer	average median time needed for questions of a topic to receive accepted answers

a topic to receive an accepted answer [10, 11]. More straightforward questions tend to receive solutions faster compared to harder questions. Measures of the difficulty metric are in Table 2.6.

The authors in [1, 2] calculated popularity and difficulty metrics for concurrency and big data topics using the five measures defined as above. Note, the values calculated for the measures dependent on questions and answers metadata and the time the information is gathered. The values for the popularity and difficulty measures for the concurrency topics are in the Table 2.7. Likewise, the values for these two measures for the big data topics are in Table 2.8.

2.3 Sentiment Analysis

To perform our sentiment analysis, we followed the following steps: For Step 1, we removed noise by preprocessing the Stackoverflow question and answers. In Step 2, we analyzed the sentiment of Stackoverflow questions and answers by using the tool Senti4SD for concurrency and big data datasets. In Step 3, we measured the sentiment for topics

Table 2.7: Popularity and Difficulty Metrics for the Concurrency Topics From the Previous Work [1]

Topic	**Popularity**			**Difficulty**	
	Avg. views	**Avg. favorites**	**Avg. scores**	**%w/o acc. answer**	**Hrs to acc. answer**
thread safety	2848	1.5	4.6	39.6	0. 3
basic concepts	2222	1.6	4.3	37.0	0.7
task parallelism	2216	1.3	4.0	35.3	0.4
locking	2152	1.3	3.5	40.1	0.3
thread life cycle management	2130	0.7	2.4	40.4	0.3
thread scheduling	2032	0.7	2.2	40.8	0.4
process life cycle management	2004	1.0	2.6	44.9	0.6
thread pool	1841	0.9	2.7	47.0	0.7
object-oriented concurrency	1773	0.8	2.6	35.2	0.3
database management systems	1727	0.6	1.8	51.2	1.0
thread sharing	1671	0.6	2.0	35.6	0.3
GUI	1664	0.5	1.6	41.1	0.4
irreproducible behavior	1647	0.6	2.3	51.1	2.1
event-based concurrency	1636	0.7	2.3	39.7	0.6
python multiprocessing	1587	0.9	2.5	50.3	0.9
entity management	1583	0.8	2.3	48.0	1.8
memory consistency	1531	1.7	4.8	33.2	0.4
file management	1458	0.6	1.9	48.8	0.6
producer-consumer concurrency	1311	0.8	2.2	43.3	0.7
unexpected output	1304	0.5	1.6	41.8	0.7
mobile concurrency	1292	0.5	1.3	50.4	0.8
runtime speedup	1276	0.9	2.7	48.0	0.7
web concurrency	1252	0.8	1.9	50.7	0.9
concurrent collections	1155	0.5	2.0	38.6	0.4
client-server concurrency	1083	0.4	1.1	50.4	0.9
data scraping	1003	0.6	1.4	48.9	1.0
parallel computing	899	0.6	1.9	50.1	2.1
Average	**1640.6**	**0.8**	**2.4**	**43.7**	**0.7**

Table 2.8: Popularity and Difficulty Metrics for the Big Data Topics From the Previous Work [2]

Topic	Popularity			Difficulty	
	Avg. views	**Avg. favorites**	**Avg. scores**	**%w/o acc. answer**	**Hrs to acc. answer**
file management	2141.6	0.5	1.4	62.3	3.7
file distribution	1861.6	0.5	1.3	63.1	6.7
hive	1811.6	0.3	1.0	66.0	3.6
dependency management	1774.7	0.4	1.4	64.2	4.7
dataframe	1731.4	0.5	1.3	46.3	1.2
string	1600.8	0.3	1.0	51.9	1.3
RDD	1543.8	0.5	1.6	49.4	1.1
hbase	1464.9	0.4	1.3	63.4	17.2
dataset load&store	1410.7	0.4	1.2	63.5	2.9
data organization	1369.3	0.8	2.0	58.5	3.2
file format	1315.1	0.5	1.2	61.9	4.0
connection management	1301.7	0.3	1.0	68.7	17.0
mapreduce model	1294.2	0.5	1.2	51.6	0.2
debugging	1278.3	0.3	1.2	64.3	2.4
basic concepts	1248.7	0.9	1.8	59.8	5.5
pyspark	1152.5	0.4	1.3	68.1	7.6
memory management	1141.1	0.6	1.7	69.0	7.7
job management	1131.2	0.4	1.5	65.7	9.2
general programming	1130.9	0.5	1.6	53.8	1.8
PIG	1079.8	0.2	0.8	63.0	10.1
date&time	1068.1	0.3	0.7	55.8	2.1
text search	1053.8	0.5	1.3	57.5	5.2
scala spark	1050.0	0.3	1.0	64.8	2.6
database import&export	1048.7	0.3	0.8	69.8	7.9
performance	840.3	0.5	1.4	64.0	3.8
logging	820.7	0.3	0.8	66.9	22.5
machine learning	763.9	0.5	1.3	61.6	4.6
stream processing	715.0	0.4	1.2	71.3	7.4
Average	**1290.8**	**0.4**	**1.2**	**61.6**	**5.9**

in the concurrency and the big data topics, which are in the Tables 2.4 and the Table 2.5, respectively. In the final step, we calculated correlations between sentiment with popularity and sentiment with difficulty using Kendall Tau bivariant correlation.

2.3.1 Preprocessing

Determining the sentiment of text with noise could lead to incorrect results. So, the text should be free of hidden metadata to perform the sentiment analysis. Consequently, we preprocessed the body of the questions and the body of the answer to reduce noise. A more thorough explanation for the preprocessing is getting a plain text, which we define as a piece of text containing alphanumeric characters.

We start by removing the code snippets that contain code using natural language processing. That is, we removed the tags </code> and </code> from the questions and answers metadata. Then, we remove all remaining HTML tags such as paragraph tags </p> and </p>, URL tags and and nbsp; extra spaces tag.

Table 2.9: Preprocessing Example

Input Text	**Processed Text**
body: <p> i use retrofit to get the response correctly, then i pass the response to an object in the response body, while it fails to get the object in, ui thread there is a nullpointerexception, error i think it's the problem of the asynchronous request, how can i avoid this problem? <p>?</p>

<pre><code> *multiple lines of code in java*
</code></pre>
 | i use retrofit to get the response correctly then i pass the response to an object in the response body while it fails to get the object in ui thread there is a nullpointerexception error i think it's the problem of asynchronous request how can i avoid this problem *(removed)* |

Furthermore, we removed any remaining newlines, extra spaces, and non-alphanumeric characters using the Java library JSoup to parse the text and convert it to a plain text.

We preprocessed 400K Stackoverflow questions and answers to plain text that belong to both concurrency and big data datasets. For instance, a question's metadata before and after preprocessing can be seen in Table 2.9. Here, the text in the left column represents the text before preprocessing, and the text in the right column gives the processed version of the same text.

2.3.2 Senti4SD

We utilize the tool Senti4SD to assess the sentiment, which can be either positive, negative, or neutral for the 400K Stackoverflow questions and answers belonging to both concurrency and big data datasets. For more details about the tool, Senti4SD refer [15].

As discussed in section 2.2.1, the 400K Stackoverflow questions and answers are an aggregate of 250K concurrency dataset and 150K big data dataset. The sentiment analysis on this dataset performed, as shown in Figure 2.1. A sample of Stackoverflow question processed by the Senti4SD tool is shown in the Table 2.10, where the input is a preprocessed text as seen in the left column, and the output sentiment assessment for this text is in the right column.

Table 2.10: A Sample Stackoverflow Question and Its Sentiment

Processed Text	**Sentiment of Text**
i use retrofit to get the response correctly then i pass the response to an object in the response body while it fails to get the object in ui thread there is a nullpointerexception error i think it's the problem of the asynchronous request how can i avoid this problem	positive

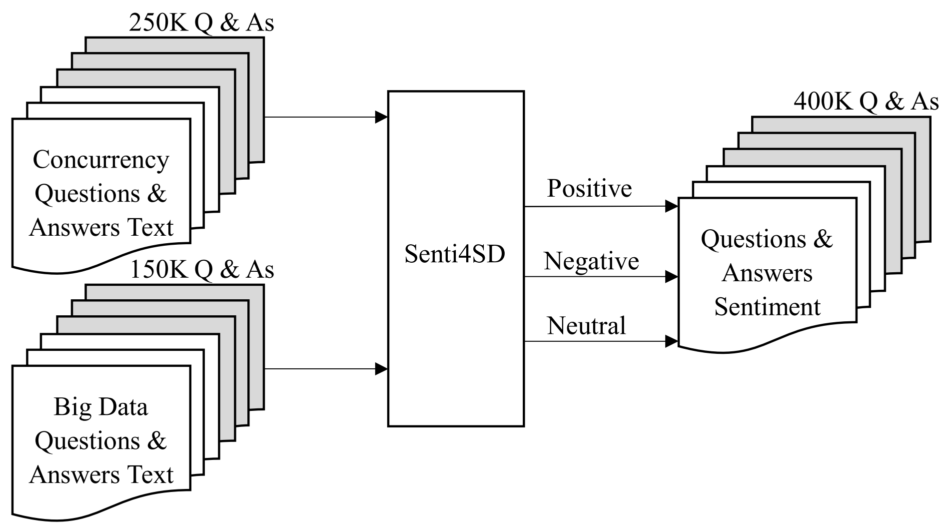


Figure 2.1: Sentiment Analysis Using Senti4SD

2.3.3 Concurrency and Big Data Topics Sentiment

In the previous section, we discussed the sentiment analysis tool Senti4SD to assess the sentiment of questions and answers belonging to concurrency and big data datasets. As the goal of our thesis is to answer the research questions ***R1*** to ***R4***, we do so by understanding the relationship between a software developer's sentiment concerning popular and difficult concurrency and big data topics. We calculate the topic sentiment by taking the mean of the sentiment assessments (positive, negative, neutral) for all the Stackoverflow questions and answers within a topic.

2.3.4 Correlations in Concurrency and Big Data

To find the correlations within the concurrency and big data datasets, first, we gathered the results from our previous step of finding sentiment for each topic within these datasets. Next, we used the popularity and difficulty metrics calculated by the authors in [1, 2]. In total, our research finds correlations between three measures of sentiment with three measures of popularity metric and three measures of sentiment with two measures of difficulty metric.

We find the correlations existing within concurrency and big data datasets using a rank-based correlation. For this, we use Kendall tau bivariant correlation, which in comparison to other rank-based correlations, is less sensitive to outliers, giving us relatively many stable results. Kendal tau computes the coefficient between two non-parametric variables belonging to the same set with different ranks. To test our hypothesis of statistical dependencies, we input our numerical results to the tool IBM SPSS [16]. A non-parametric test means that the distribution is without any underlying assumptions. In short, it means that the distribution has no prior knowledge about the probabilities of the variables being assessed.

By evaluating the correlations given by the tool, we answer our research questions ***R1*** to ***R4***. The flow of our analysis is in Figure 2.2. In the next chapter, we discuss our results in depth.

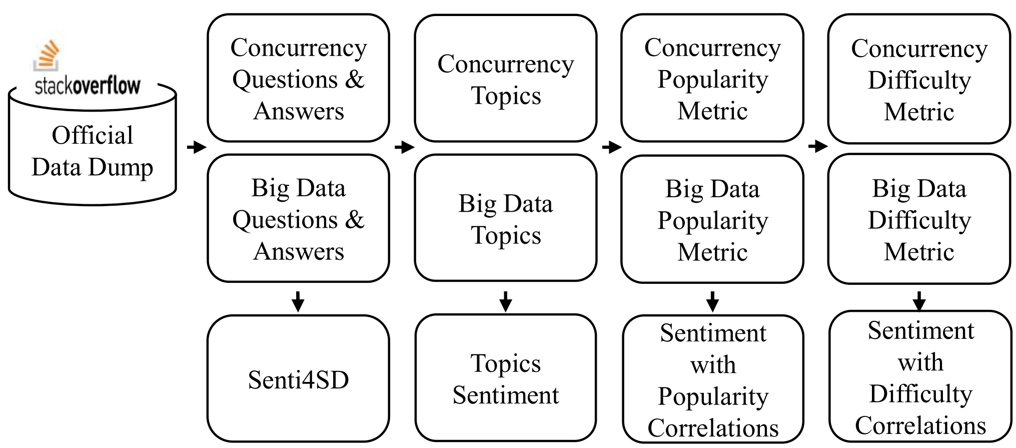


Figure 2.2: Methodology

CHAPTER THREE

RESULTS

In this chapter, we present the discussion of our research findings for concurrency and big data datasets. First, we present the results of our sentiment analysis. Next, we analyze the correlations. Finally, we conclude this chapter by answering our research questions ***R1*** to ***R4*** with a detailed analysis of the results for both concurrency and big data datasets.

3.1 Concurrency and Big Data Sentiment

In this section, we present our results for concurrency and big data dataset topics sentiment.

Initially, we calculate the sentiment percentages for each topic within the concurrency dataset, which we presented in Table 2.4. We compute the sentiment percentages for the concurrency topics by evaluating the sentiment for all the questions and answers and take the mean of sentiments measure (positive, negative, neutral) for a topic.

The results for positive, negative, neutral sentiment percentages regarding the concurrency dataset, are shown in Table 3.1. For instance, from the table, if we pick the positive polarity, we find that the *data scraping* topic has the highest rate of positive polarity. In contrast, *process life cycle management* topic has the lowest positive polarity of Stackoverflow questions and answers. The *process life cycle management* keeps the lowest sentiment percentage in negative polarity, and *irreproducible behavior* topic claims the highest negative polarity. The same two topics, *process life cycle management* and *irreprod- ucible behavior*, switch their polarities from lowest to highest, highest to lowest respectively in neutral polarities. Overall, concurrency sentiment analysis concludes by evaluating the mean sentiment for all the topics, which are 16.1%, 23.4%, and 60.2% to positive, negative, and neutral sentiment.

Similar to the calculations we did for the concurrency, we compute the big data topics sentiment percentages. The result of our calculations is in Table 2.5. The resulting mean of sentiment measures (positive, negative, neutral) for each topic belonging to the big data dataset is in Table 2.5.

In the big data topic, *basic concepts* topic has the highest rate of positive polarity, and *scala spark* topic has less positively polarized Stackoverflow questions and answers. These two topics switch their polarities from high to low and vice versa for negative polarity. The topic *general programming* has a higher percentage of questions and answers with neutral polarity, whereas *debugging* topic has the lowest. Overall big data sentiment analysis concludes by evaluating the mean sentiment for all the topics, which are 16.8%, 36.4%, and 46.6% to positive, negative, and neutral sentiment, respectively.

3.2 Concurrency and Big Data Sentiment Correlations

To understand the role of the sentiment towards software developers' productivity, we find the correlations existing between sentiment with popularity and sentiment with difficulty. To do so, we gather the sentiment for each topic within concurrency and big data datasets calculated in the previous section. Then we consider the popularity and difficulty metrics computed by the authors [1, 2], as in Table 2.7 and the Table 2.8 for concurrency and big data, respectively. Finally, using the tool IBM SPSS, we find the correlations existing in concurrency and big data datasets to answer our research questions ***R1*** to ***R4***.

We discussed the IBM SPSS tool to find the correlations. Using this methodology, the results of our correlations for the concurrency dataset are shown in the Table 3.3. In our computations, we found a negative correlation exists between the popularity metric's average favorites measure and negative sentiment with a correlation significance at 0.001.

Table 3.1: Concurrency Topics Sentiment

Topic	**Positive**	**Negative**	**Neutral**
thread safety	20.4	19.3	60.1
basic concepts	19.3	20.9	59.7
task parallelism	13.5	20.0	66.3
locking	9.6	18.6	71.6
thread life cycle management	10.7	22.6	66.5
thread scheduling	14.3	28.4	57.1
process life cycle management	7.6	14.0	78.3
thread pool	11.6	16.6	71.7
object-oriented concurrency	16.4	20.1	63.4
database management systems	21.7	21.4	56.8
thread sharing	10.5	24.5	64.9
GUI	16.7	30.7	52.4
irreproducible behavior	16.3	43.0	40.6
event-based concurrency	11.7	20.2	67.9
python multiprocessing	17.3	26.9	55.6
entity management	21.5	20.1	58.2
memory consistency	15.0	16.8	68.1
file management	12.1	18.5	69.3
producer-consumer concurrency	14.4	23.1	62.4
unexpected output	18.3	36.9	44.7
mobile concurrency	19.4	31.1	49.4
runtime speedup	16.6	20.9	62.4
web concurrency	18.2	22.6	59.1
concurrent collections	20.3	16.8	62.8
client-server concurrency	16.9	30.8	52.1
data scraping	26.7	27.7	45.5
parallel computing	18.8	20.8	60.3
Average	**16.1**	**23.4**	**60.2**

Table 3.2: Big Data Topics Sentiment

Topic	**Positive**	**Negative**	**Neutral**
file management	11.3	28.9	59.7
file distribution	15.1	38.4	46.3
hive	20.6	36.7	42.6
dependency management	12.3	48.5	39.0
dataframe	20.7	27.9	51.3
string	19.6	23.7	56.5
RDD	16.2	30.6	53.1
hbase	22.4	29.2	48.2
dataset load&store	16.9	35.3	47.7
data organization	14.2	27.6	58.0
file format	15.3	29.3	55.3
connection management	15.9	43.8	40.2
mapreduce model	21.3	26.4	52.2
debugging	15.8	54.2	29.9
basic concepts	25.2	23.3	51.3
pyspark	13.1	51.1	35.7
memory management	15.2	38.4	46.3
job management	11.8	41.6	46.4
general programming	14.0	24.1	61.7
PIG	16.2	43.3	40.4
date&time	20.8	29.2	49.8
text search	18.1	32.1	49.7
scala spark	6.4	54.8	38.6
database import&export	18.4	43.5	37.9
performance	20.9	36.6	42.3
logging	17.1	40.1	42.6
machine learning	20.1	38.6	41.2
stream processing	15.6	43.0	41.2
Average	**16.8**	**36.4**	**46.6**

Table 3.3: Correlations in Concurrency

Metric	Measure	Positive	Negative	Neutral
popularity	average views	-/.01849	-/.14996	+/.07294
	average favorites	-/.19815	-/.00154	+/.00889
	average scores	+/.07911	-/.00127	+/.00320
difficulty	% w/o acc. answer	+/.08344	+/.02289	-/.00758
	hrs. to acc. answer	+/.00731	+/.08687	-/.00565

This means that, if the questions and answers within a concurrency topic are tilted towards the negative end of the emotions scale, then the users marking these questions and answers as their favorite are fewer. Also, there exists a positive correlation between the popularity metric's average scores measure and neutral sentiment with a significance at 0.003, which means that the neutrally toned questions and answers receive higher scores by the users.

Table 3.4: Correlations in Big Data

Metric	Measure	Positive	Negative	Neutral
popularity	average views	-/.60734	-/.13308	.04802
	average favorites	+/.96703	-.00292	+/.00406
	average scores	-/.22997	-/.07308	.04983
difficulty	% w/o acc. answer	-/.13308	+/.00006	-/.00009
	hrs. to acc. answer	-/.63265	.00908	-/.01964

Using the same approach in handling the big data dataset, the results of the correlations we computed were presented in Table 3.4. Our popularity results indicate that there is a negative and positive correlation between negative and neutral sentiment, respectively, with respect to the average favorites. On the other hand, our difficulty metric results point that there exists a positive and negative correlation between negative and neutral sentiment, respectively, with the percentage of questions with an accepted answer measure. At this point in our analysis, we have the results to answer our research questions.

3.3 Discussion

We answer the research questions ***R1*** to ***R4*** introduced at the beginning of this thesis, using the correlations in concurrency and big data datasets, which can be found in the Tables 3.3 and the Table 3.4. The summary of these correlations are in the Tables 3.5 and in the Table 3.6.

Table 3.5: Summary of the Concurrency Sentiment Correlations

Sentiment	**Metric**	**Direction**	**Measure**
positive	popularity		no correlation
	difficulty		no correlation
negative	popularity	negative	average favorites
		negative	average scores
	difficulty		no correlation
neutral	popularity	positive	average scores
	difficulty	positive	hrs. to acc. answer

R1. Is there a relation between sentiment with popularity and sentiment with difficulty in concurrency topics? The summary of the popularity and difficulty metrics and their correlations with sentiment are displayed in Table 3.5. For the negative and neutral sentiment, we observe correlations between both popularity and difficulty metrics. However, we did not find any correlations between positive sentiment with both popularity and difficulty metrics.

R2. Is there a relation between sentiment with popularity and sentiment with difficulty in big data topics? The summary of the popularity and difficulty metrics and their correlations with sentiment are displayed in Table 3.6. For the negative and neutral sentiment, we observe correlations between both popularity and difficulty metrics. No correlation was found between positive sentiment with popularity and difficulty measures. Likewise, during our concurrency dataset analysis, we did not find correlations between positive sentiment with popularity and difficulty.

Table 3.6: Summary of the Big Data Correlations

Sentiment	**Metric**	**Direction**	**Measure**
positive	popularity		no correlation
	difficulty		no correlation
negative	popularity	negative	average favorites
	difficulty	positive	% w/o acc. answer
neutral	popularity	positive	average favorites
	difficulty	positive	% w/o acc. answer

R3. What sentiment do concurrency and big data developers express towards a common topic? The concurrency and big data sentiment values are displayed in the Table 3.3 and Table 3.4 respectively. The *basic concepts* is a common topic in both concurrency and big data topics. The questions and answers discussed in this topic are generic to the concurrency and big data topics. *Basic concepts* topic has a common property, which is that the questions and answers within these topics are more neutrally toned in both concurrency and big data datasets.

R4. What commonality do the correlations hold toward concurrency and big data? Referring to the Tables 3.5 and 3.6, the commonly maintained property that our computations lead towards a negative sentiment which is inversely correlated with the average favorites measure in popularity metric.

CHAPTER FOUR

IMPLICATIONS

Studying the sentiment correlations in concurrency and big data datasets, we investigate the impact of our findings on different communities. The following are the implications we deduced using the research questions ***R1*** to ***R4***, regarding specific communities: software practitioners, researchers, and educators. We abbreviated the implications for practitioner's, researcher's and educator's as PI, RI, and EI respectively.

4.1 Practitioners

PI.1 By referring to the sentiment correlations in the Table 3.5 and Table 3.6 considering our research findings for question ***R4***. We observed that concurrency and big data developers are sensitive towards negative and neutral sentiment compared to the positive sentiment.

Using the directions observed for negative sentiment with popularity metric, we deduce that concurrency developers' average favorites and average scores measure decreases when they detect more negative sentiment in the Stackoverflow questions and answers. Similarly, big data developers' average favorites measure decreases with the negative sentiment in the Stackoverflow questions and answers. Again, using the directions for neutral sentiment with popularity metric, we deduce that concurrency developers' average scores measure rises when a neutral sentiment is detected in the Stackoverflow questions and answers. Likewise, for the big data developers, average favorites measure increases with a neutral sentiment. The difficulty metrics sentiment correlations show that the concurrency developers take longer to receive an accepted answer when they detect neutral sentiment. On the other hand, big data developers will have fewer Stackoverflow questions with accepted answers when they detect both negative and neutral sentiment.

PI.2 Our findings for research answers to ***R1*** and ***R2*** implies that the concurrency and the big data developers receive fewer average favorites measure when Stackoverflow questions and answers have a negative sentiment.

PI.3 We observed from research answer ***R3*** that concurrency and big data developers commonly express neutral sentiment towards basic concepts topic.

4.2 Researchers

RI.1 Our research findings imply correlations between negative and neutral sentiment with popularity and difficulty metrics, refer to Table 3.5 and Table 3.6. However, our research did not find a correlation between positive sentiment with popularity and difficulty with both concurrency and big data topics. Researchers can further explore the sentiment and its effects on various other metrics, not limited to popularity and difficulty. This aggregated research data to understand the human emotions coefficient can further help evolving technology that primarily depends on people.

4.3 Educators

EI.1 Educators in their practice can refer to the utilization of the Stackoverflow platform while solving popular and difficult topics, especially with concurrency and big data topics. Considering our findings for research question ***R1*** and the sentiment correlations in the Table 3.5, educators, can help concurrency developers recognize the role of negative and neutral sentiment with average favorites, average scores and hours to accepted answer measures. Similarly, our findings for research question ***R2*** and the sentiment correlations in Table 3.6, educators can assist big data developers in understanding the role of negative and neutral sentiment to receive average favorites and percentage of questions with accepted answer measures.

CHAPTER FIVE

RELATED WORK

In this section, we discuss the works that are related to our current thesis, which perform sentiment analysis.

Guzman et al. [17] focuses on the human factor by analyzing the sentiment of 60,425 commit logs. These commit logs belong to 90 of the top starred software projects on Github. For these 90 projects, they studied the relationship between sentiment with three areas, which are programming languages, team distribution, and project approvals. Their research findings conclude that commit logs for Java projects tend to be more negative, projects with more distributed teams tend to have more positive commit logs, and commit comments on Mondays are more negatively toned.

Sinha et al. [18] expanded the data to analyze the sentiment of 2M non-empty commit logs from 28K projects using BOA [19] tool. Their research concludes that the majority of the commit logs on GitHub projects are neutral. An interesting observation is, the projects commit logs that comprise 10% more negative emotion in comparison with positive sentiment. For some popular projects, Wednesday and Thursday commit logs are most negatively toned. Finally, a strong positive correlation exists with the sentiment in the commit logs, and the number of files changed in the commits.

In the previous research, Ortu et al. [20] took a different approach to perform sentiment analysis. Their study focused on human "affectiveness" and developers' productivity in the Agile environment. For their analysis, they performed an empirical analysis on more than 560K comments from 14 Apache projects using the Jira issue tracking system. For the "affectiveness" metric, they concentrated on emotion, sentiment, and politeness. The research conveyed that emotions reflect on software developers' problem-solving productivity. By which, developers who post joyful comments take a shorter time to fix a problem compared to the developers with gloomy comments take longer to fix a similar problem. Additionally, they analyze politeness and its complex role in the developer's productivity.

Souza and Silva [21] analyzed the sentiment of the commit logs belonging to the Travis CI continuous integration server. Their dataset includes 1,262 Github projects. Concluding that the build process can both be "affects" and "affected" by negative sentiment, and developers writing continuous integration servers are more positive while writing commit messages.

Garcia et al. [5], performed sentiment analysis on the open-source project GENTOO using the bug tracking platform BUGZILLA, to study the relation between emotions and activity of contributors. The study finds that whenever strong positive emotions or negative emotions or when an unexpected emotional deviation occurs, contributors are likely to abandon the project.

Bazelli et al. [22], this study explores the personality traits of the Stackoverflow users by studying the relations between the sentiment of the developer's answers and their reputation. This study finds that the top-rated users express strong positive emotions in their answers compared to medium and low ranked users and that users with a higher number of favorites posts express more positive emotion than users with a lower number of favorite posts.

Muriga et al. [23], studied if issue reports carry any emotional information. Their dataset uses 117 open source projects belonging to Apache software. This study confirms that issue reports reportedly have emotions that can affect design choices, maintenance activity, and colleagues.

Rahman et al. [24], studies automated code comment generation by analyzing 292 Stackoverflow code fragments and 5K discussion comments. The study finds that with high precision and recall, insightful comments can be extracted. 80% of auto-generated comments are accurate, precise, concise, and useful for the participants.

Zhang and Hou [25], proposes ways to extract problematic API features using online discussion forums, and for this analysis, they use sentiment analysis to determine the polarity of posted sentences. This study mainly focuses on the negative sentiment for their analysis.

Similarly, Goul et al. [26], used sentiment analysis to determine the usefulness and interaction of employees while utilizing business intelligence apps. Panichella et al. [27], uses sentiment analysis to determine apps usability, apps that are available on Google Play, and Apple Stores by checking users' comments and categorizing the comments as software maintenance, evolution, bug tracking and more.

The primary resource that we use in this study to perform our sentiment analysis is Ahmed and Bagherzadeh [1] for concurrency dataset. For the big data dataset, we used Bagherzadeh and Khatchadourian [2]. Specifically, it evaluates topic modeling, popularity, and difficulty metrics and their correlations.

CHAPTER SIX

CONCLUSION AND FUTURE WORKS

We have discussed our large scale sentiment analysis in this thesis. In this final chapter, we present future work and conclusion.

6.1 Conclusion

We performed sentiment analysis on a large scale dataset with 400K Stackoverflow questions and answers belonging to concurrency and big data datasets in comparison to the related work.

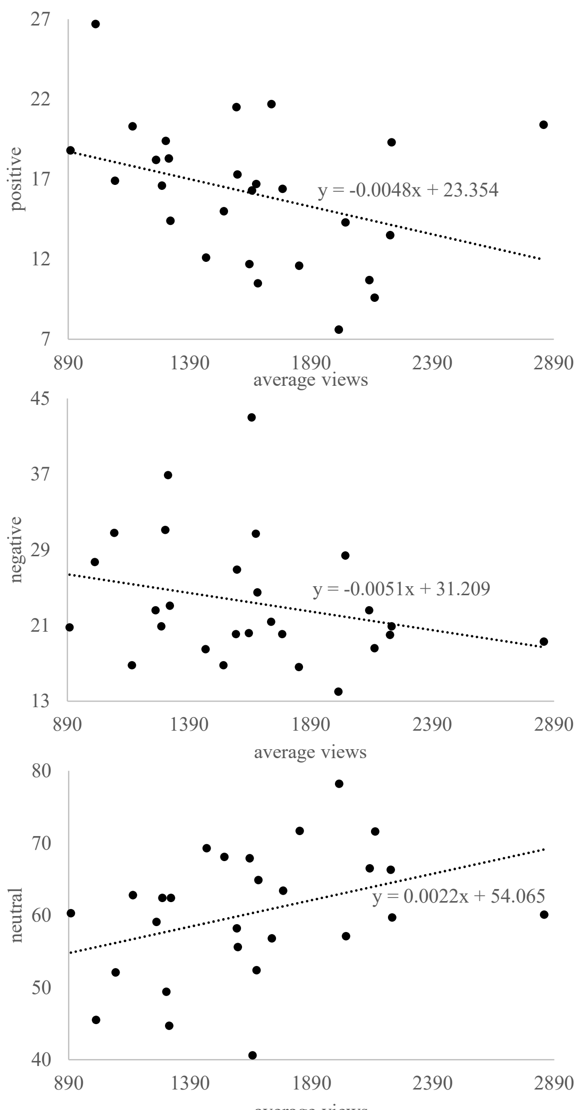
We found a strong inverse correlation between the popularity metric average favorites measure with the negative sentiment in both concurrency and big data datasets. Indicating that concurrency and big data software developers expressing less negative emotions tend to gain more likes from the users. Concurrency and big data correlations demonstrate that negative and neutral sentiment has some correlations with popularity and difficulty metrics. We also found that concurrency and big data developers are not sensitive towards positive sentiment, as we did not find correlations between positive sentiment with popular- ity and difficulty. We conclude that the software developer's sentiment towards a challeng- ing task has an impact on their productivity.

6.2 Future Works

We plan to explore the sentiment of software practitioners developing different technologies like security, programming languages, and deep learning.

In this section, we provided the scattered plots for the correlations between sentiment with popularity and sentiment with difficulty, belonging to both concurrency and big data.

Figure .1: Popularity Average Views Measure With Sentiment Correlations in Concurrency

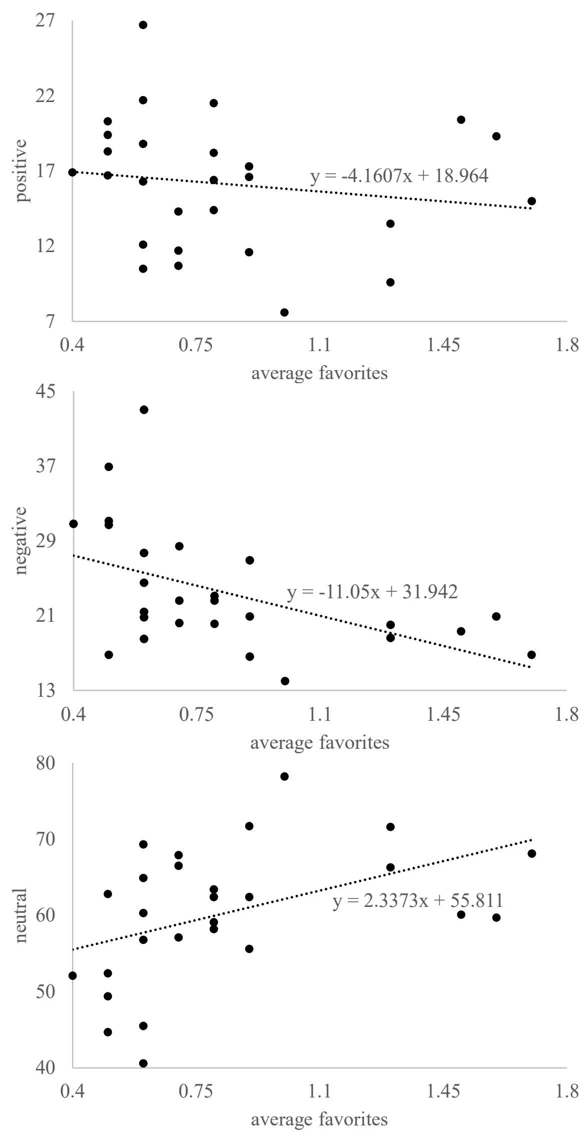


The average views measures' relation with the positive sentiment is an inverse relationship. No significant correlations found, however.

The average views measure keeps an inverse relationship with the negative sentiment. Again, no significant correlations found between them.

The average views measure has a direct relation with the neutral sentiment, no indication of existing correlations also.

Figure .2: Popularity Average Favorites Measure With Sentiment Correlations in Concurrency

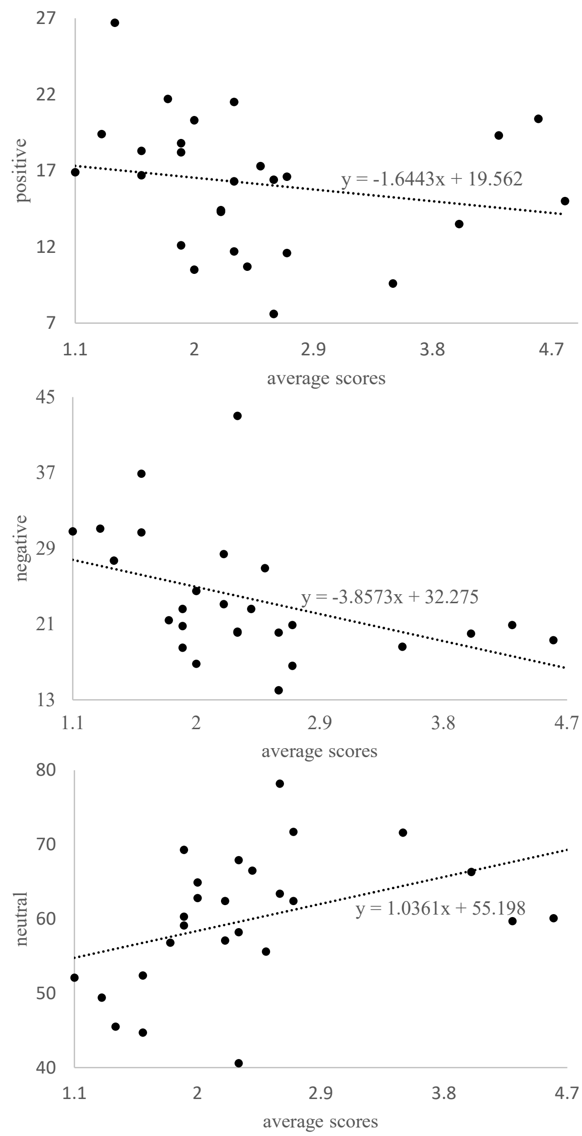


The average favorites measure has an inverse relationship with the positive sentiment. We did not find a significant correlation between them.

The average favorites measure has an inverse relationship with the negative sentiment, we found a correlation significance at 0.001.

The average favorites measure has a direct correlation with the neutral sentiment, and no correlations significance found.

Figure .3: Popularity Average Scores Measure With Sentiment Correlations in Concurrency

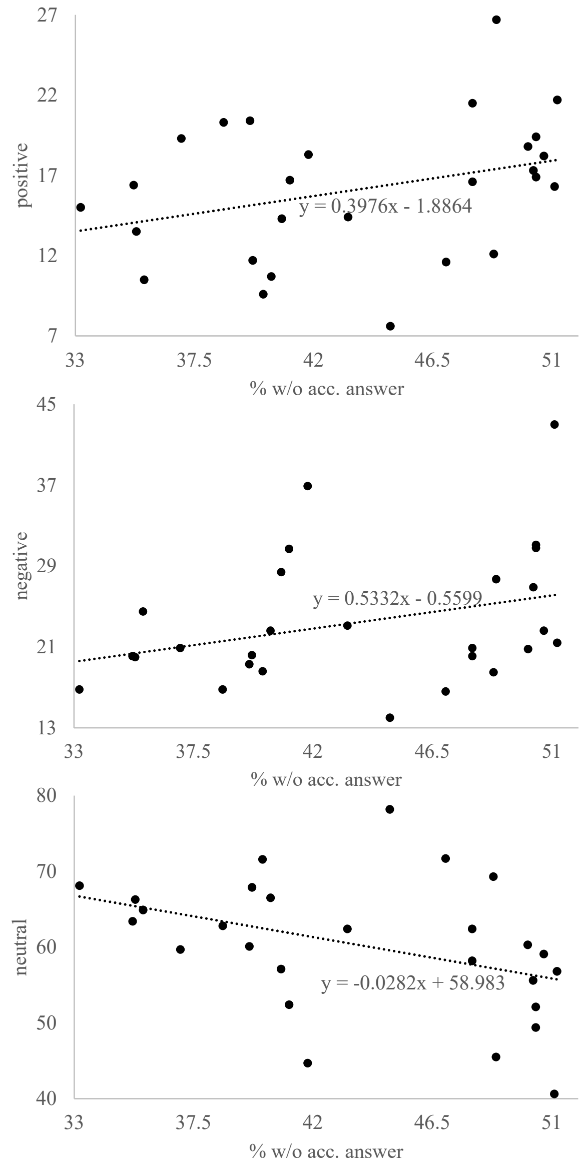


The average scores measure has in an inverse relationship with the positive sentiment, no correlations found.

The average scores measure has an inverses relationship with the negative sentiment, with a correlation significance at 0.001.

The average score measure has a direct relation with the neutral sentiment, with a significant correlation at 0.003.

Figure .4: Difficulty %W/o Acc. Answer Measure With Sentiment Correlations in Concurrency

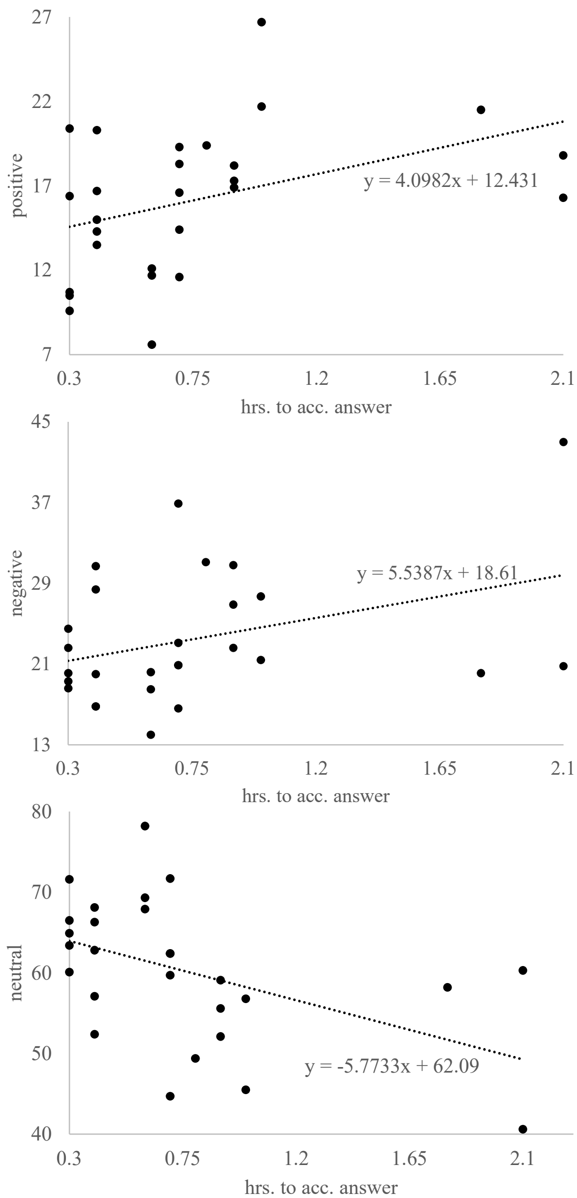


The %w/o acc. answer measure with a direct relationship with the positive sentiment has no significant correlation.

The %w/o acc. answer measure with a direct relationship with the negative sentiment has no significant correlation.

The %w/o acc. answer measure with a direct relationship with the negative sentiment has no significant correlation.

Figure .5: Difficulty Hrs. to Acc. Answer Measure With Sentiment Correlations in Concurrency

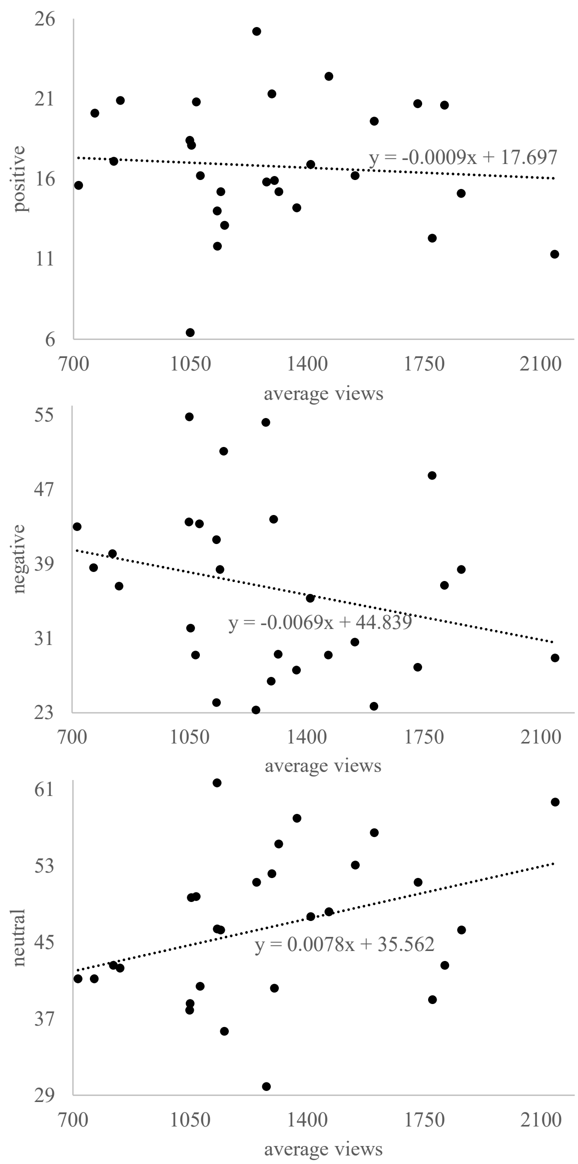


The hrs. to acc. answer measure with a direct relationship with the positive sentiment has no significant correlation.

The hrs. to acc. answer measure with an inverse relationship with the negative sentiment has no significant correlation.

The hrs. to acc. answer measure with a direct relationship with the neutral sentiment, has a significant correlation of 0.005.

Figure .6: Popularity Average Views Measure With Sentiment Correlations in Big Data

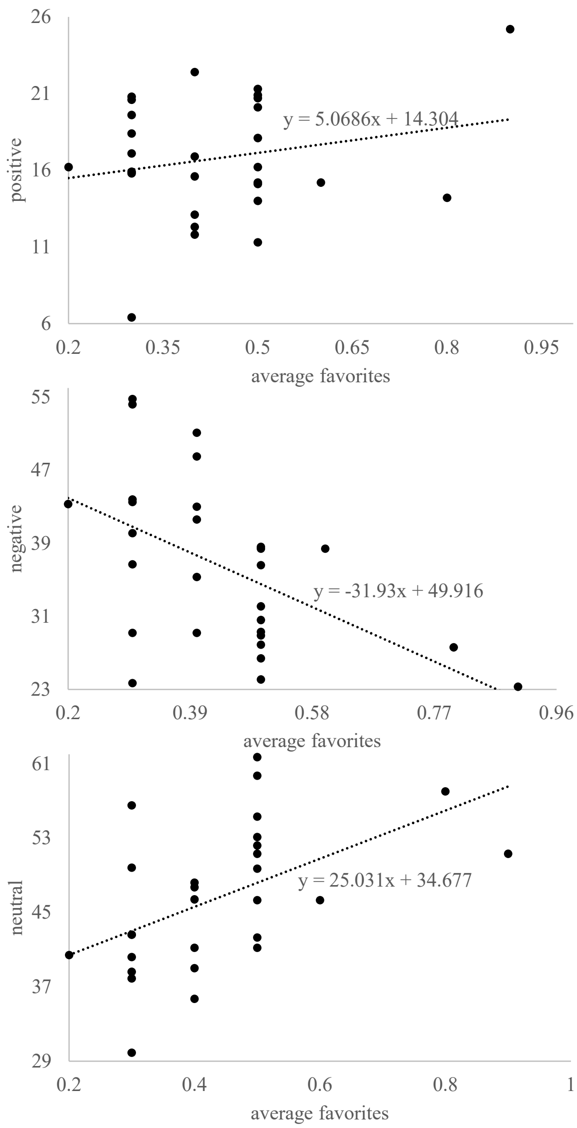


The average views measure has an inverse relation with the positive sentiment, we did not find significant correlation.

Similarly, average views measure maintain an inverse relation with the negative sentiment, we did not find significant correlation.

The average views measure has a direct relation with the neutral sentiment, we did not find significant correlation.

Figure .7: Popularity Average Favorites Measure With Sentiment Correlations in Big Data

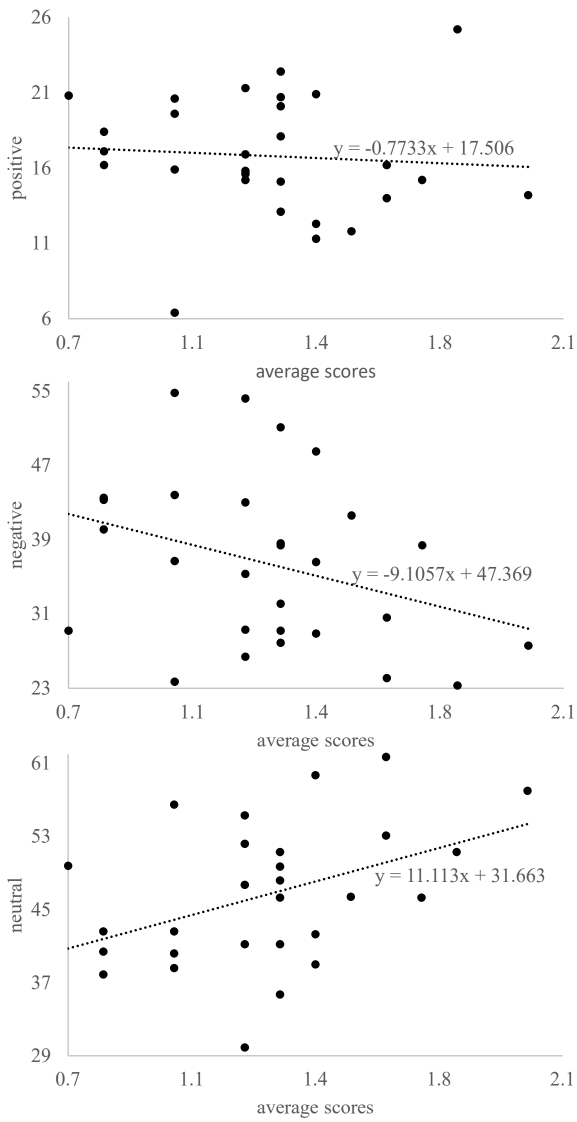


The average favorites measure has a direct relation with the positive sentiment, we did not find significant correlation.

The average favorites measure has an inverse relation with the negative sentiment, and at 0.002 significant correlation.

The average favorites measure has a direct relation with the neutral sentiment, and at 0.004 significant correlation.

Figure .8: Popularity Average Scores Measure With Sentiment Correlations in Big Data

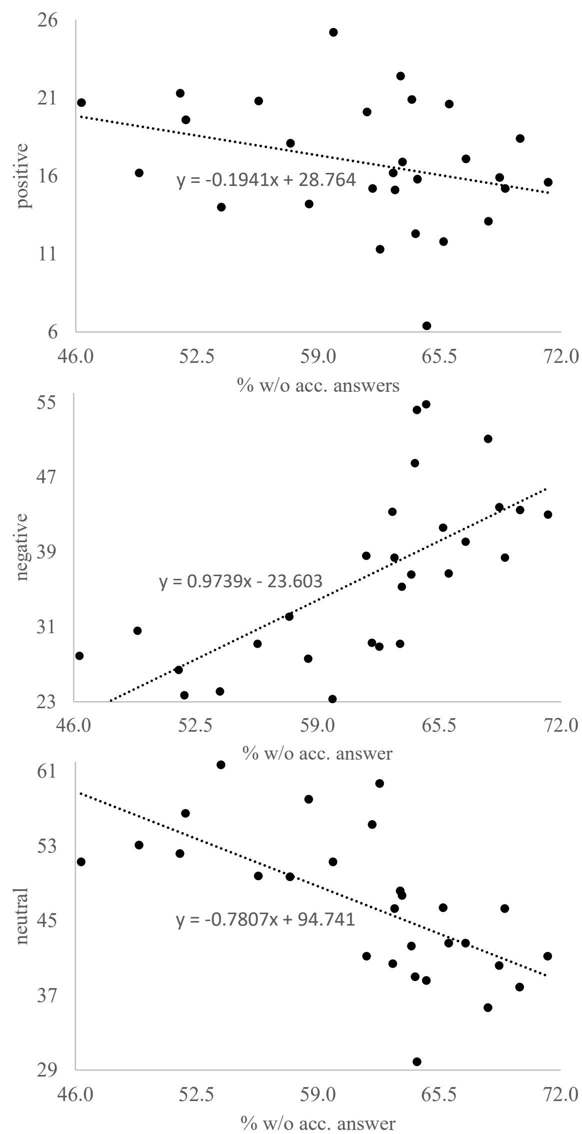


The average scores measure has an inverse correlations with the postive sentiment, we did not find significant correlation.

Similarly, average scores measure has an inverse correlations with the postive sentiment, we did not find significant correlation.

The average scores measure has a direct correlations with the neutral sentiment, we did not find significant correlation.

Figure .9: Difficulty %W/o Acc. Answer Measure With Sentiment Correlations in Big Data

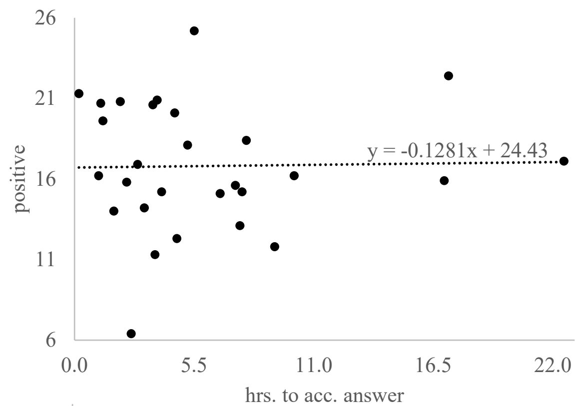


The %w/o acc. answer measure has an inverse relations has no correlations with the positive sentiment.

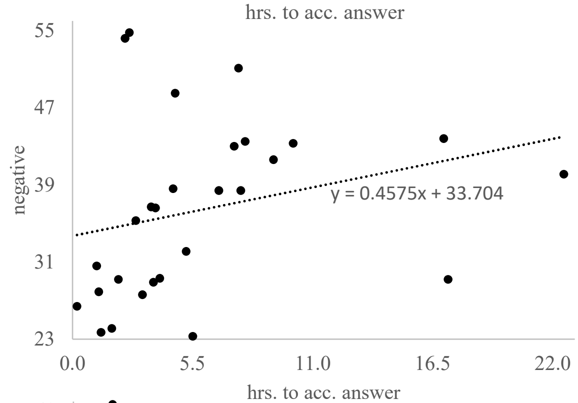
The %w/o acc. answer measure has a direct relations has a significant correlations at 0.00006 with the negative sentiment.

The %w/o acc. answer measure has a direct relations has a significant correlations at 0.00009 with the neutral sentiment.

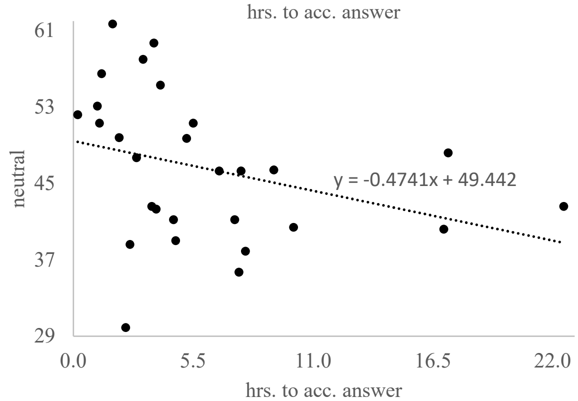
Figure .10: Difficulty Hrs. to Acc. Answer Measure With Sentiment Correlations in Big Data



The hrs. to acc. answer measure has an inverse relation and no correlations found with the positive sentiment.



The hrs. to acc. answer measure has a direct relation and no correlations found with the negative sentiment.



The hrs. to acc. answer measure has an inverse relation and no correlations found with the neutral sentiment.

REFERENCES

[1] S. Ahmed and M. Bagherzadeh, "What Do Concurrency Developers Ask About?A Large-Scale Study Using Stack Overflow," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 30, ACM, 2018.

[2] M. Bagherzadeh and R. Khatchadourian, "Going Big: A Large-Scale Study on What Big Data Developers Ask," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 432–442, ACM, 2019.

[3] D. Graziotin, X. Wang, and P. Abrahamsson, "Happy Software Developers Solve Problems Better: Psychological Measurements in Empirical Software Engineering," *PeerJ*, vol. 2, p. e289, 2014.

[4] B. W. Boehm and P. N. Papaccio, "Understanding and Controlling Software Costs," *IEEE transactions on software engineering*, vol. 14, no. 10, pp. 1462–1477, 1988.

[5] D. Garcia, M. S. Zanetti, and F. Schweitzer, "The Role of Emotions in Contributors Activity: A Case Study of the GENTOO Community," in *2013 International Conference on Cloud and Green Computing*, pp. 410–417, IEEE, 2013.

[6] "Stack Exchange Dump." `https://stackexchange.com/sites?view=list#traffic`, 2019. oct.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[8] K. Bajaj, K. Pattabiraman, and A. Mesbah, "Mining Questions Asked by Web Developers," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, pp. 112–121, 2014.

[9] S. Nadi, S. Krüger, M. Mezini, and E. Bodden, "Jumping Through Hoops: Why Do Java Developers Struggle With Cryptography APIs?," in *Proceedings of the 38th International Conference on Software Engineering*, pp. 935–946, 2016.

[10] C. Rosen and E. Shihab, "What Are Mobile Developers Asking About? A Large Scale Study Using Stack Overflow," *Empirical Software Engineering*, vol. 21, no. 3, pp. 1192–1223, 2016.

[11] X.-L. Yang, D. Lo, X. Xia, Z.-Y. Wan, and J.-L. Sun, "What Security Questions Do Developers Ask? A Large-Scale Study of Stack Overflow Posts," *Journal of Computer Science and Technology*, vol. 31, no. 5, pp. 910–924, 2016.

[12] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design Lessons From the Fastest QA Site in the West," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 2857–2866, 2011.

[13] G. Pinto, W. Torres, and F. Castor, "A Study on the Most Popular Questions About Concurrent Programming," in *Proceedings of the 6th Workshop on Evaluation and Usability of Programming Languages and Tools*, pp. 39–46, 2015.

[14] C. Treude, O. Barzilay, and M.-A. Storey, "How Do Programmers Ask and Answer Questions on the Web?," in *Proceedings of the 33rd international conference on software engineering*, pp. 804–807, 2011.

[15] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Sentiment Polarity Detection for Software Development," *Empirical Software Engineering*, vol. 23, no. 3, pp. 1352–1382, 2018.

[16] "IBM." `https://www.ibm.com/analytics/academic-statistical-software`, 2019. oct.

[17] E. Guzman, D. Azócar, and Y. Li, "Sentiment Analysis of Commit Comments in GitHub: An Empirical Study," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, pp. 352–355, ACM, 2014.

[18] V. Sinha, A. Lazar, and B. Sharif, "Analyzing Developer Sentiment in Commit Logs," in *Proceedings of the 13th International Conference on Mining Software Repositories*, pp. 520–523, ACM, 2016.

[19] Dyer, Robert and Nguyen, Hoan Anh and Rajan, Hridesh and Nguyen, Tien N., "Boa: A language and infrastructure for analyzing ultra-large-scale software repositories," in *35th International Conference on Software Engineering*, ICSE'13, pp. 422–431, May 2013.

[20] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli, "Are Bullies More Productive? Empirical Study of Affectiveness Vs. Issue Fixing Time," in *Proceedings of the 12th Working Conference on Mining Software Repositories*, pp. 303–313, IEEE Press, 2015.

[21] R. Souza and B. Silva, "Sentiment Analysis of Travis CI Builds," in *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, pp. 459–462, IEEE, 2017.

[22] B. Bazelli, A. Hindle, and E. Stroulia, "On the Personality Traits of StackOverflow Users," in *2013 IEEE international conference on software maintenance*, pp. 460–463, IEEE, 2013.

[23] A. Murgia, P. Tourani, B. Adams, and M. Ortu, "Do Developers Feel Emotions? An Exploratory Analysis of Emotions in Software Artifacts," in *Proceedings of the 11th working conference on mining software repositories*, pp. 262–271, ACM, 2014.

[24] M. M. Rahman, C. K. Roy, and I. Keivanloo, "Recommending Insightful Comments for Source Code Using Crowdsourced Knowledge," in *2015 IEEE 15th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pp. 81–90, IEEE, 2015.

[25] Y. Zhang and D. Hou, "Extracting Problematic API Features From Forum Discussions," in *2013 21st International Conference on Program Comprehension (ICPC)*, pp. 142–151, IEEE, 2013.

[26] M. Goul, O. Marjanovic, S. Baxley, and K. Vizecky, "Managing the Enterprise Business Intelligence App Store: Sentiment Analysis Supported Requirements Engineering," in *2012 45th Hawaii International Conference on System Sciences*, pp. 4168–4177, IEEE, 2012.

[27] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "How Can I Improve My App? Classifying User Reviews for Software Maintenance and Evolution," in *2015 IEEE international conference on software maintenance and evolution (ICSME)*, pp. 281–290, IEEE, 2015.