Analyzing Developer Sentiment in Commit Logs

Vinayak Sinha, Alina Lazar, Bonita Sharif
Department of Computer Science and Information Systems
Youngstown State University
Youngstown, Ohio USA 44555
vsinha@student.ysu.edu, {alazar, bsharif}@ysu.edu

ABSTRACT

The paper presents an analysis of developer commit logs for GitHub projects. In particular, developer sentiment in commits is analyzed across 28,466 projects within a seven year time frame. We use the Boa infrastructure's online query system to generate commit logs as well as files that were changed during the commit. We analyze the commits in three categories: large, medium, and small based on the number of commits using a sentiment analysis tool. In addition, we also group the data based on the day of week the commit was made and map the sentiment to the file change history to determine if there was any correlation. Although a majority of the sentiment was neutral, the negative sentiment was about 10% more than the positive sentiment overall. Tuesdays seem to have the most negative sentiment overall. In addition, we do find a strong correlation between the number of files changed and the sentiment expressed by the commits the files were part of. Future work and implications of these results are discussed.

CCS Concepts

• Software and its engineering → Software organization and properties → Software system structures → Ultra-large-scale systems • Information systems → Data mining

Keywords

sentiment analysis; commit logs; Java projects

1. INTRODUCTION

There is an increasing amount of research in the software engineering community dealing with sentiment and the emotional aspect of software development. Sentiment analysis or opinion mining was initially developed as an automated method of extracting sentiment polarity from short texts posted online such as movie reviews, product reviews, microblogs and tweets. Recently, this method was adopted by the software engineering community and applied to different software engineering artifacts such as commit logs [1], question and answer posts and online mailing list messages [2]. The sentiments a developer projects during development are important as they could have an impact on productivity. The more we understand about developer emotions, the better we can support them by providing better tools for them during development.

There has been some prior work in this area. Guzman et al. analyzed 60.425 commit messages [1] from 90 top rated GitHub projects. They found Java projects tend to have more negative comments with distributed teams having more positive comments. They also found that Mondays had the most negative emotion associated with them. Murgia et al. conduct an exploratory study to have humans rate or agree on emotions in issue reports [3]. They found that developers do indeed express emotions and positive emotions had higher agreements between human raters. The goal was to eventually automate emotion mining in software artifacts. Jongeling et al. provide a good comparison of four sentiment analysis tools for software engineering research and also conduct an analysis of whether the tool sentiment matches a human evaluator's sentiment [4]. They found that the tools gave contradictory results when run on issue tracker data. They call for more tools targeting software engineering artifacts.

The work presented in this paper resembles the work by Guzman et al. [1] with some important differences. First, we analyze developer sentiment in commit logs on a much larger set – 2,251,585 commit logs. Second, we also take a look at the number of files changed and map them to the sentiment expressed in the commits that the files were part of. We do this across the entire project's lifetime up until 2015. In this paper, we seek to answer the following research questions.

RQ1: What is the general developer sentiment in commit messages for GitHub projects?

RQ2: What is the relationship between developer sentiment in commit messages and the day of the week the commit was made?

RQ3: Is there a correlation between the number of changed files and developer sentiment?

We first describe the dataset used. The sentiment analysis tool that is used on the commit logs is described in Section 3. Section 4 describes the results to our research questions. Finally, we discuss our results, and state our conclusions and future work.

2. DATASET USED

We wrote *Boa* scripts that we ran through the web-based Boa interface. This allowed us to download all the commit logs from the GitHub Medium (September 2015) dataset provided for the MSR 2016 challenge [5]. This was the dataset used to answer the research questions posed above. Each commit log can be uniquely identified by the project id and the revision id. After eliminating the empty commit messages, a total of 2,251,585 non-empty commit messages remained in the dataset. The commit logs belong to 28,466 projects.

The GitHub projects available in *Boa* under the GitHub Medium (September 2015) dataset have creation dates between 2007-2013. Therefore, while investigating RQ3 only commits with submission dates between 2007 and 2013 were taken into account. After removing all other commits the final dataset contains 2,130,474 commit logs. We provide all the *Boa* scripts used to retrieve the data at *http://seresl.csis.ysu.edu/MSR16challenge* along with other related supplementary material.

MSR'16, May 14-15, 2016, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/2901739.2903501

3. SENTIMENT ANALYSIS APPROACH

To determine the sentiment polarity developers convey while submitting code revisions and commit logs, we use the sentiment analysis tool SentiStrength [6]. This tool was chosen because of the high accuracy rates reported in previous studies on Twitter data. SentiStrength was also used in software engineering studies. For example Guzman et al. [1] investigated the relationship between sentiment in commit messages and the programming language used, the day of the week when the commit was submitted and the overall project approval.

First, SentiStrength tokenizes the text and then assigns a score for each word that conveys an emotion. Words with negative implications are given scores between -1 to -5 and words with positive sentiment values receive an integer value between 1 and 5. Next, the word's scores for each commit log are summarized to generate a pair of values known as the *senti score*. The first value in the *senti score* indicates the positive score and second value is the negative score for the sentence. Let us illustrate this with an example. Consider the following commit log: "Added basic flying monster animation in project" taken from project ID "10002651". After analyzing each word SentiStrength provides the scores for each word and the sentence as "Added[0] basic[0] flying[0] monster[-1] animation[0] [[Sentence=-2,1=word max, 1-5]][[[1,-2 max of sentences]]]". Note the *senti score* of [1,-2] where 1 indicates the maximum positive score for the sentence and -2 is the maximum negative score for the sentence. To find the final sentiment score for a commit log, we take the sum of the maximum positive and the maximum negative score given by the *senti score*. The *final sentiment score* is used to find the overall commit log polarity sentiment as positive, neutral or negative. In the above example, the final sentiment score would be -1 (sum of 1 and -2). A positive sum represents a positive sentiment, zero represents a neutral sentiment or no emotion, and a negative sum indicates a negative sentiment. Table 1 shows examples from all three categories of commits from our GitHub dataset and the final score that is a sum of the *senti score*.

Guzman et al. [1] count as positive any commit log with a positive score (1,5]. Similarly, they count as negative any commit log with a negative score [-5,1). In this way, a commit log could be counted as positive and negative. We use the approach presented in Jongeling et al. [4] and calculate the final score as a sum of the maximum positive and the maximum negative score that SentiStrength provides.

4. RESULTS

We present the results for each of our research questions posed in the Introduction in the following sections.

4.1 RQ1: General Sentiment in Commit Logs

The results of running SentiStrength on all 2,251,585 commit logs are given in Table 2. We notice that 74.74% of the commits had a neutral sentiment, 7.19% had a positive sentiment and 18.05% had a negative sentiment. This indicates that there were more than twice the number of negative sentiment commit logs compared to positive sentiment commit logs. We notice that a maximum number of the commits fall into the *senti score* range of [1,-1]. This could happen because many of the commit messages have URLs and/or variable names in them rendering them as neutral. This is also indication that further work is needed to adapt sentiment analysis tools to software engineering artifacts.

Table 1. Positive, Negative, and Neutral Commit Logs

Sentiment	Commit Message	Final Score
Positive	We're not totally terrible.	4
	Build success !!!	3
	pretty pretty code	3
	Added parallelism and seems it works fine :)	3
	A few finishing touches that Anna liked :)	3
	Small tweaks on top of Daniel's excellent refactoring git-svn-id	3
Negative	Terrible, terrible mock folder guid retrieval.	-4
	Trying to complete the qualifier 3. Grounds for suicide :(	-4
	Fix heinous TMemoryBuffer bug and warning in FileTransport Review	-4
	Attempted to fix map camera failed horribly	-4
	ENH: very painfully merge: svn merge --accept	-4
Neutral	initial commit Committer: Jeremy Truelove jtruelove@gmail.com	0

To analyze this further, we split the dataset into three subsets: large, average, and low number of commits and considered only the top five projects in each of these categories.

Table 2. Sentiment across all commits

Sentiment	Final Sentiment Score	Number of Commits	Sentiment Percentage
Negative	-4	66	18.053%
	-3	2793	
	-2	39770	
	-1	363853	
Neutral	0	1683009	74.748%
Positive	1	149931	7.199%
	2	11782	
	3	371	
	4	10	
	Total	**2251585**	

Table 3 shows the three split datasets for further analysis. The split was done manually after looking at a sorted distribution of commits in the projects. A total of 83,936 commits were part of the subset analysis.

Table 3. Top five projects from the subset of data categorized into large, average, and low number of commits

Data Subset	Total # Commits	Min Commits	Max Commits	Mean Commits
Large	54471	9360	14969	10894.2
Average	23240	4574	4746	4648
Low	6225	1235	1254	1245

Table 4. Sentiment in projects with large, average and low number of commits.

	Number of Commits		
Sentiment	**Large**	**Average**	**Low**
Negative	21.14%	22.33%	11.49%
Neutral	71.05%	70.45%	82.47%
Positive	7.81%	7.22%	6.04%

We report the results of the sentiment score in Table 4 after running SentiStrength on these subsets. We notice similar trends in these subsets when we compare them to Table 2. The only difference is that for projects with low number of commits, the positive and negative sentiment seem to fall closer together (only 5% apart) whereas in projects with large and average number of commits, the negative sentiment is on average 14% higher than the positive sentiment.

This could be because as the project progresses (with more commits), it gets more complex involving more developers and thus more issues arise causing sentiment to move towards the negative direction. This also does not necessarily mean that the project is not productive or of good quality. Another set of analysis needs to be conducted to determine code quality, which in turn needs to be mapped to the sentiment analysis done here.

One reason for the high neutral sentiment could be because commits in general are different than tweets or online reviews. When people write reviews their goal is to express feelings of satisfaction or dissatisfaction about a product or movie. Software developers write commit logs anytime a revision is submitted to a software repository, therefore most of the time there is no human emotion or sentiment involved. The small percentage of commits that exhibit a positive or a negative sentiment polarity show different types of emotions than other types of online postings [3].

4.2 RQ2: Sentiment by Day of Week

In this research question, we wanted to determine if the day of the week plays a role in developer sentiment for commit logs. We removed all commits with a commit date before their project's creation date and all commits with a date in the future. All the projects that remained were created between 2007 and 2013. There were 2 projects created in 2007 and 416,812 projects created in 2013. Based on the commit date, we calculated the day of the week the commit was made and grouped sentiment by day.

For this analysis, we choose to look at two representative scores given by SentiStrength namely, the maximum positive with the least negative sentiment (MAX +ve) and the maximum negative with the least positive sentiment (MAX –ve). We believe these two extremes are more important as a lot of the sentiment gets averaged out if scores are added together. Figure 1 shows the percentage of these sentiment scores across the day of week along with the percentage of commits done on that day. It can be seen that most commits were done on Wednesday, which also sees the second highest maximum positive and maximum negative sentiment. The highest maximum negative sentiment is seen on Tuesday across all projects. Considering only weekdays, the lowest positive sentiment and the lowest negative sentiment are both seen on Mondays with 5% more positive sentiment.

Figure 2 shows a further breakdown of these percentages across three categories of commits in Table 3. For projects with the most number of commits (far left of the figure) we see the highest negative sentiment on Wednesday and Thursday. Thursday also had the highest positive sentiment (and lowest negative sentiment). In the large subset, SentiStrength only had values for Wednesday and Thursday for the maximum negative sentiment.

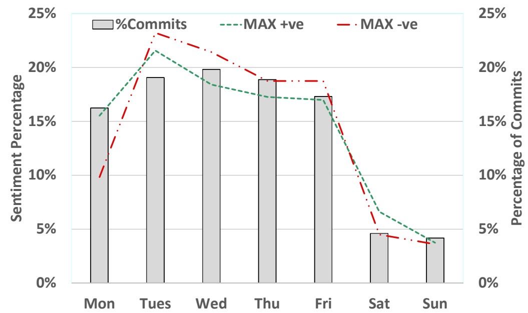

Figure 1. Maximum positive and maximum negative sentiment across all projects with respect to day of week

Hence this group of large committers do not follow the average of all projects as shown in Figure 1 where in general Tuesday was the most negative day. The projects with average number of commits (middle chart in Figure 2) had Tuesday as the most negative but we also find that it had the most positive sentiment. Finally, Tuesday was again the day with the most negative sentiment and Fridays had the most positive sentiment for the projects with fewer commits (low category). To conclude RQ2 findings, we find that there are trends in sentiment across the days of the week that differ based on the project's number of commits.

4.3 RQ3: Sentiment and Number of Files Changed

In RQ3, we wanted to determine if there was any relationship between the number of changed files in a commit and sentiment seen in commit logs. To do this, we queried Boa to give us all the files that were added, modified, and deleted across the top five projects in each of the large, average, and low commit categories. The number of files changed is the sum of all the files that were added, modified, and deleted. See Figure 3 for the results. We group together final scores of positive, negative, and neutral sentiment to show how sentiment changes across time along with the number of files changed. The number of files changed (line graphs) in a commit mapped to each sentiment is shown on the Y-axis to the right. The Y-axis on the left denotes the average number of changed files per commit (bar graph) during the year.

For the *large* subset of the top 5 projects, we notice that in the year 2014 there was a maximum number of changed files per commit (~60.35). We do notice a spike in the negative sentiment at this time as well (see 2010 for similar trend). There is a spike in positive sentiment too but not as prominent as the negative sentiment. In the *average* subset, we find 2009 and 2011 to be the most negative overall. The year 2011 also has the highest number of changed files. However, we also see a decrease in negative sentiment in Year 2010 from 2009 even though the number of files changes was higher in 2010. In the *low* category of commits, the negative and positive sentiment are almost the same across all the years. There is an unusual spike in neutral sentiment in the year 2014. The maximum number of files changed was in the year 2012 which caused the positive and the negative sentiment to spike slightly in the following year.

We found strong correlations (using Pearson's correlation test > 0.95) between the negative, positive, and neutral number of commits and the average number of changed files especially for the large and low subsets but not the average subset.

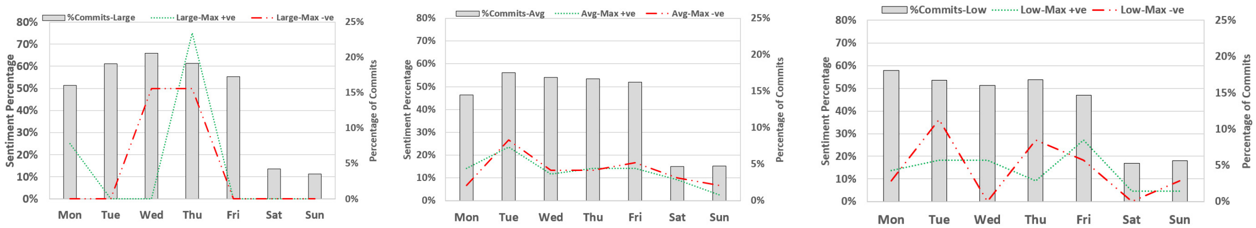

Figure 2. Maximum positive and maximum negative sentiment in top 5 projects with large, average, and low commits.

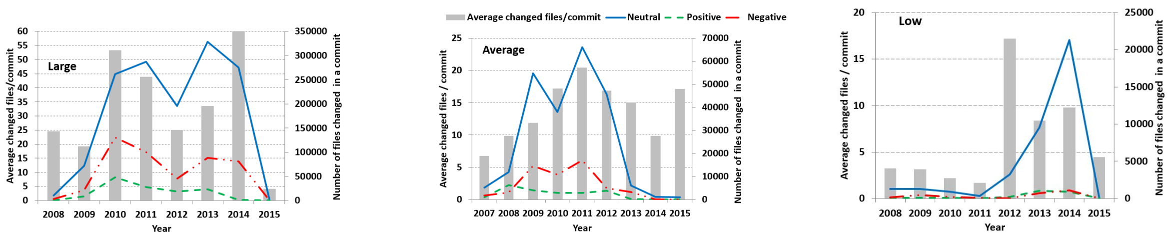

Figure 3. Sentiment and files changed over time in top 5 projects with large, average, and low commits.

5. DISCUSSION

As with any real world data, we found the commit logs to contain some questionable values. We found dates that stem from 1970 as well as dates that were in the future such as 2025. These were removed during the analysis of RQ2 and RQ3. Comparing our results of RQ1 to tweets [7] we find that our GitHub commits have 18% (from Table 2) negative sentiment, tweets about scientific papers and tweets about agile project management tools had 1% and 11% negative sentiment respectively. The positive sentiment is 7.199%, 4.20%, and 42% for our analysis of GitHub commits, tweets on scientific papers, and tweets about agile project management tools respectively. Clearly, a lot more negative sentiment is expressed in GitHub commit logs when compared to twitter logs.

Guzman et al. [1] also look into the sentiment of developers by day of week (RQ2). They report that commit logs submitted on Monday have a more negative emotion than commits submitted on any other working day of the week. We conclude that Tuesday was the most negative day overall for all commits. Since we used different datasets than [1], we can't necessarily compare these results directly.

Our results for RQ3 provide strong correlation between the number of files changed and the sentiment carried by the commit that contained the files. However, more work is needed in this area to clearly understand how this relationship impacts the project as a whole. We believe our results are a first step in this direction. We also found more negative sentiment in prior years than more recent years which could be indicative of project stability.

6. CONCLUSIONS AND FUTURE WORK

The paper presents a study of sentiment analysis on GitHub commit logs. We found that a majority of the sentiment in GitHub projects are categorized as neutral but when comparing positive with negative sentiment, we found more than twice the percentage of negative sentiment than positive ones when analyzing all the commit logs in the specified dataset. Overall, more negative sentiment was detected on Tuesday, however, for the top five projects with the most commits, Wednesdays and Thursdays were the most negative. Finally, there is positive correlation with sentiment and the number of files changed. Future work can look into the specific type of file change (such as an addition, deletion or modification of a file) to determine if any relationship exists. In the future, we plan on replicating this study using the GitHub large dataset and other sentiment analysis tools. We will also work towards training sentiment tools on a validated set of commit messages to make them more robust for software engineering problems.

7. REFERENCES

[1] 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*, 2014, pp. 352–355.

[2] P. Tourani, Y. Jiang, and B. Adams, "Monitoring sentiment in open source mailing lists-exploratory study on the apache ecosystem," in *Proceedings of the 2014 Conference of the Center for Advanced Studies on Collaborative Research (CASCON), Toronto, ON, Canada*, 2014, pp. 74–95.

[3] 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*, 2014, pp. 262–271.

[4] R. Jongeling, S. Datta, and A. Serebrenik, "Choosing your weapons: On sentiment analysis tools for software engineering research," in *Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference on*, 2015, pp. 531–535.

[5] R. Dyer, H. A. Nguyen, H. Rajan, and T. N. Nguyen, "Boa: A language and infrastructure for analyzing ultra-large-scale software repositories," in *Proceedings of the 2013 International Conference on Software Engineering*, 2013, pp. 422–431.

[6] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *J. Am. Soc. Inf. Sci. Technol.*, vol. 61, no. 12, pp. 2544–2558, Dec. 2010.

[7] N. Friedrich, T. D. Bowman, W. G. Stock, and S. Haustein, "Adapting sentiment analysis for tweets linking to scientific papers," *ArXiv Prepr. ArXiv150701967*, 2015.