Sentiment Analysis of issues in GitHub

Yaoxian Su

1. Introduction

Issues comment events are the most important user behavior events [1]. They are also rich source of information carrying the full range of emotions [2]. Sentiment analysis is always an important measurement of emotions and satisfaction which severely impact the quality and efficiency of issues.

My project aims to analyze the relationship between lexical sentiment and time characteristics of GitHub issues. I apply fastText algorithm to get the sentiment scores of selected GitHub issues and analyze their statistical characteristics. Then I study the following topics based on sentiment scores and created/closed time of issues:

1) Are emotions in issue comments related to the day of the week in which the commits were written?

2) Are emotions in issue comments related to their created time of the day?

3) Are emotions in issue comments related to the time span between the created date and the closed date?

2. Related work

1) Sentiment analysis of commit comments in GitHub: an empirical study

Emitza Guzman et al. [3] apply lexical sentiment analysis to study emotions expressed

in commit comments of different open source projects. The dataset is provided by Gousios [4] containing a total 60425 commit comments.

They use SentiStrenth, a lexical sentiment extraction tool specialized in dealing with short, low quality texts, to assigning a quantitative mood value (positive or negative) to a text snippet, followed by statistical methods to get its emotion score. Then they applied Wilcoxon rank sum test and the Spearman correlation to find their relationship with used programming language, the day of week or time in which the commits were written team geographical distribution and project approval.

2) Towards emotional awareness in software development teams

Emitza Guzman and Bernd Bruegge [5] propose an approach to improve emotional awareness in software development teams by means of quantitative emotion summaries.

They use latent Dirichlet allocation (LDA) [6] for extracting topical information from the artifacts and lexical sentiment analysis [7] for assigning the emotion score, and interview the project leaders of the three teams to evaluate the usefulness of their approach.

3) Bag of Tricks for Efficient Text Classification

Armand Joulin et al. [8] represent a new baseline model called fastText which can extremely reduce time cost but maintain accuracy.

They evaluate this model on sentiment analysis and tag prediction. For the first evaluation, fatsText outperforms other algorithms while goes up to at least a 15,000× speed-up.

4) An exploratory qualitative and quantitative analysis of emotions in issue report comments of open source systems

Alessandro Murgia et al. [9] demonstrate the feasibility of a machine learning classifier for identifying issue comments.The analysis of the issue comments is based on 117 projects of the Apache Software Foundation.

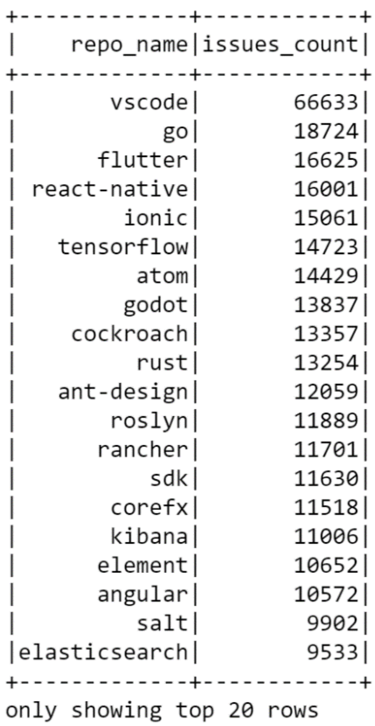
They construct a machine learning classifier specifically trained to identify emotions in issue comments. For each classifier, they construct five variants based on different, popular classification algorithms: Support Vector Machine (SVM), Naive Bayes (NB), Single Layer Perceptron (SLP), K-Nearest Neighbor (KNN) and Random Forest (RF), and then evaluate them according to the accuracy, precision, recall, F1 and AUC.

3. Experiments

3.1. GitHub data collection

All issue data are collected from the GitHub Archive and obtained by writing a python script that calls GitHub REST API. Firstly, I compose the query on Google BigQuery to get the 500-top repositories containing the most issue comment events in githubarchive of 2018. However, by checking the details of these repositories, I realize some repositories have been transferred or deleted, and some are not about programming projects. After sorting them by stargazers count and checking programming language and existence of issues, I select top 250 projects as my research projects in this work.

Then I recursively request to download all issues in selected repositories by GitHub REST API for issues. It’s noticeable that as far as the API is concerned, every pull request is an issue but not every issue is a pull request, which means this endpoint may also return pull requests in the response. If an issue is a pull request, the object will include a pull request key: "pull_request", and the body doesn’t contain any useful text. 11 repositories that only have pull request issues are useless in this work. After filtering issues by checking pull request key, created time and closed time, I get all textual issues from 2017.1 to 2019.4, including 1064458 issue comments in total. Table 1 shows top 20 picked repositories with most issue counts.

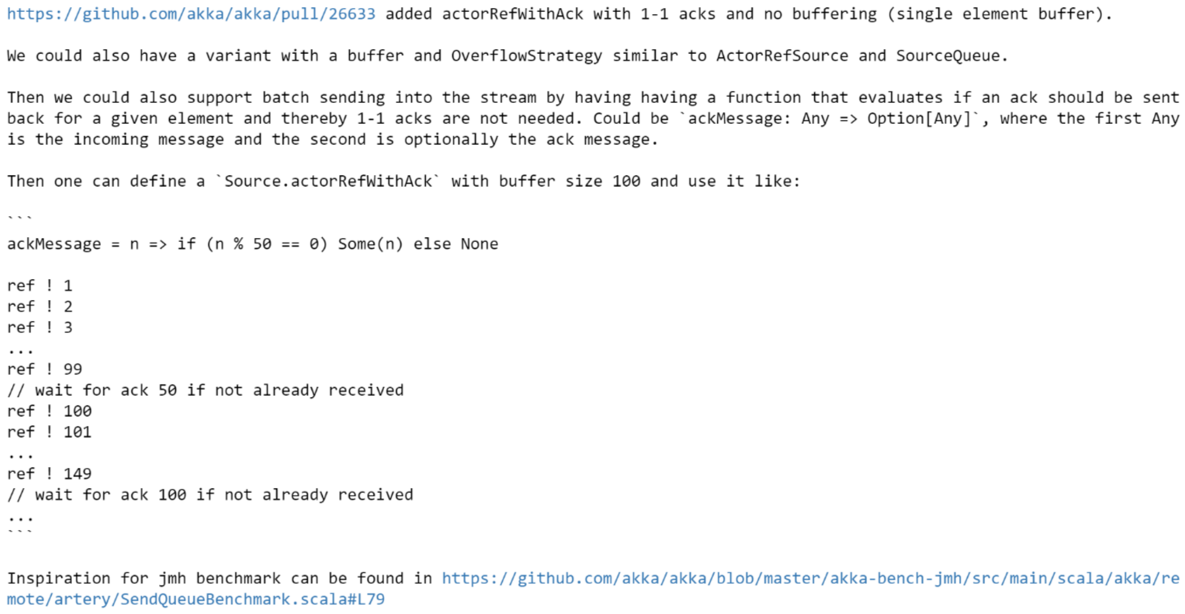


```
+-------------+------------+
|    repo_name|issues_count|
+-------------+------------+
|       vscode|       66633|
|           go|       18724|
|      flutter|       16625|
| react-native|       16001|
|        ionic|       15061|
|   tensorflow|       14723|
|         atom|       14429|
|        godot|       13837|
|    cockroach|       13357|
|         rust|       13254|
|   ant-design|       12059|
|       roslyn|       11889|
|      rancher|       11701|
|          sdk|       11630|
|       corefx|       11518|
|       kibana|       11006|
|      element|       10652|
|      angular|       10572|
|         salt|        9902|
|elasticsearch|        9533|
+-------------+------------+
only showing top 20 rows
```

Table 1. picked repositories and their issue counts.

3.2. Data exploration and processing

Generally, GitHub issues contains not only discussions about repositories, but also some codes. And by checking the text body of issues, I notice that there are some special characters that need cleaning: 1) @user; 2) #hashtag; 3) “’ codes’’’; 4) <!--- system message ---!>; 5) web links; 6) some unicodes; 7) punctuations. Below is an example of text body.



~~~
https://github.com/akka/akka/pull/26633 added actorRefWithAck with 1-1 acks and no buffering (single element buffer).

We could also have a variant with a buffer and OverflowStrategy similar to ActorRefSource and SourceQueue.

Then we could also support batch sending into the stream by having having a function that evaluates if an ack should be sent
back for a given element and thereby 1-1 acks are not needed. Could be `ackMessage: Any => Option[Any]`, where the first Any
is the incoming message and the second is optionally the ack message.

Then one can define a `Source.actorRefWithAck` with buffer size 100 and use it like:

```
ackMessage = n => if (n % 50 == 0) Some(n) else None

ref ! 1
ref ! 2
ref ! 3
...
ref ! 99
// wait for ack 50 if not already received
ref ! 100
ref ! 101
...
ref ! 149
// wait for ack 100 if not already received
...
```

Inspiration for jmh benchmark can be found in https://github.com/akka/akka/blob/master/akka-bench-jmh/src/main/scala/akka/re
mote/artery/SendQueueBenchmark.scala#L79
~~~

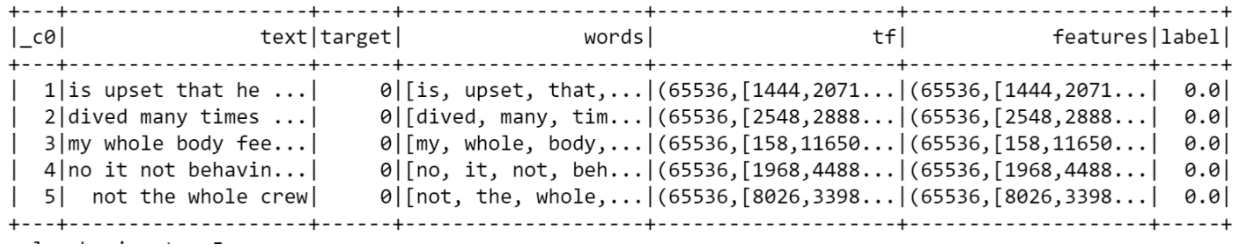
Figure 1. an example of issue body.

The Pyspark function ‘regexp_replace’ is used here to clear these signs and contents inside. Besides, converting all characters to lowercase and stopwords removing are also needed here, and null rows are dropped.

3.3. Sentiment analysis

The training and validation data are generated from the sentiment140 dataset* from Stanford University. The dataset contains the polarity of the tweet (0 = negative, 1 = positive or neutral) and the texts of 1,600,000 tweet. After NLP pre-processing, 90% of them are used as training data, and others are divided as validation set. They also follow the same pre-processing described in the former section.

After these texts are tokenized, their Term Frequency-Inverse Document Frequency (TF-IDF) is calculated as the text mining feature. The logistic regression model from pyspark.ml.classification package is applied to predict sentiment scores. Below is an example of training results.



```
+---+--------------------+------+--------------------+--------------------+--------------------+-----+
|_c0|                text|target|               words|                  tf|            features|label|
+---+--------------------+------+--------------------+--------------------+--------------------+-----+
|  1|is upset that he ...|     0|[is, upset, that,...|(65536,[1444,2071...|(65536,[1444,2071...|  0.0|
|  2|dived many times ...|     0|[dived, many, tim...|(65536,[2548,2888...|(65536,[2548,2888...|  0.0|
|  3|my whole body fee...|     0|[my, whole, body,...|(65536,[158,11650...|(65536,[158,11650...|  0.0|
|  4|no it not behavin...|     0|[no, it, not, beh...|(65536,[1968,4488...|(65536,[1968,4488...|  0.0|
|  5|  not the whole crew|     0|[not, the, whole,...|(65536,[8026,3398...|(65536,[8026,3398...|  0.0|
+---+--------------------+------+--------------------+--------------------+--------------------+-----+
```

Table 2. an example of training results.

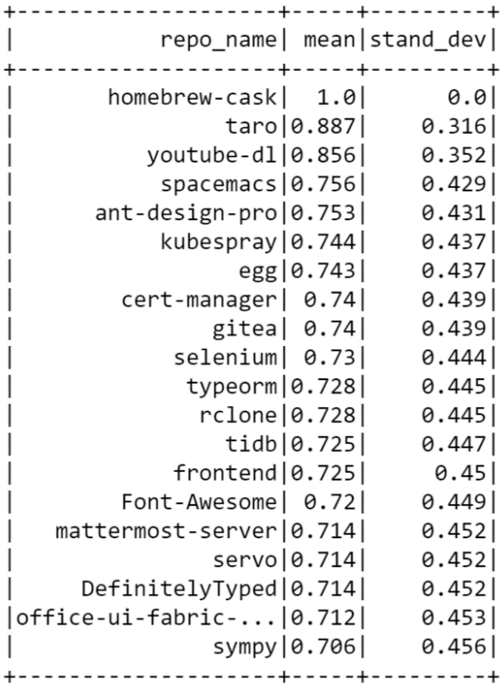
The ROC-AUC of validation achieves 85.82%. Then the same methodology is applied to pre-processed issue texts in terms of testing.

4. Results

In the following sections, the analysis of the emotions expressed in issue comments relation between and their relationship with additional factors such as time of the day, date of the week, time span between created and closed dates.

1) Emotions in Issue comments

The average emotion score of the issue comments for each of the projects tended to neutrality, with mean at 2.038. This is because many of the GitHub comments only describe technical aspects without emotions presented, or are only slightly positive or negative [3]. Table 3 shows the top 20 projects with the emotion average score and standard deviation for each one of them.



```
+--------------------+-----+---------+
|           repo_name| mean|stand_dev|
+--------------------+-----+---------+
|       homebrew-cask|  1.0|      0.0|
|                taro|0.887|    0.316|
|          youtube-dl|0.856|    0.352|
|           spacemacs|0.756|    0.429|
|      ant-design-pro|0.753|    0.431|
|           kubespray|0.744|    0.437|
|                 egg|0.743|    0.437|
|        cert-manager| 0.74|    0.439|
|               gitea| 0.74|    0.439|
|            selenium| 0.73|    0.444|
|             typeorm|0.728|    0.445|
|              rclone|0.728|    0.445|
|                tidb|0.725|    0.447|
|            frontend|0.725|     0.45|
|        Font-Awesome| 0.72|    0.449|
|   mattermost-server|0.714|    0.452|
|               servo|0.714|    0.452|
|      DefinitelyTyped|0.714|    0.452|
|office-ui-fabric-...|0.712|    0.453|
|               sympy|0.706|    0.456|
+--------------------+-----+---------+
```

Table 3. the top 20 projects with their emotion score averages respectively.

While the average emotion score of the issue comments tended to be slightly positive (the overall mean is 0.583), there are some highly emotional issues and other positive and negative issues whose emotional content is not reflected in the average. In order to further examine these contents, the proportion of positive and negative issues is worth analysis. Figure 2 shows the bar chart of all projects and their mean sentiment scores. Figure 3 shows this distribution for all projects with their positive and negative issue comments. Figure 4 is the box plot of these means. From them we can see that most of projects contains neutral mean sentiment scores in issues, while others appear to be more positive. And most of projects have similar distribution of positive and negative issue comments, and others consist of more positive issues.

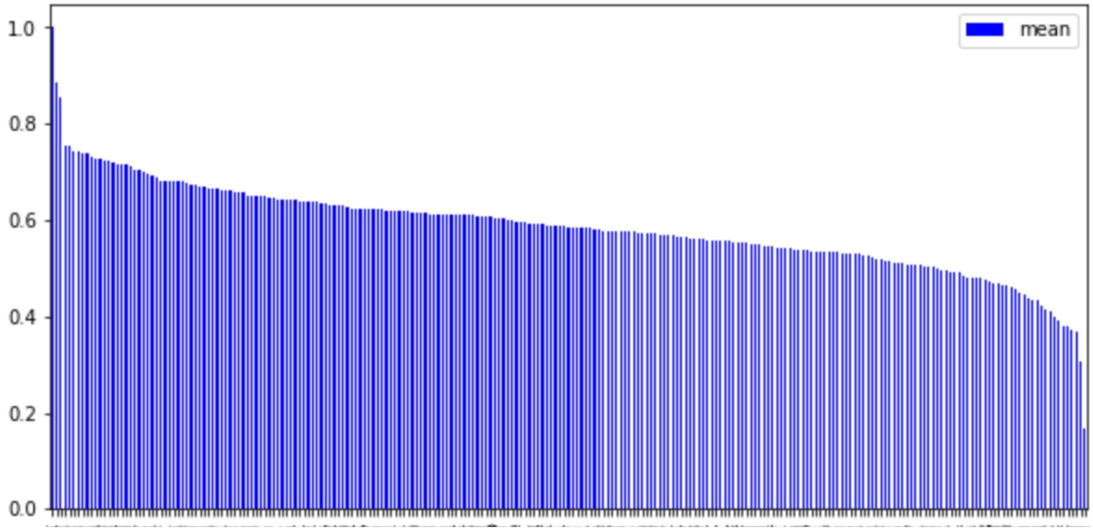


Figure 2. Emotion score average per project.

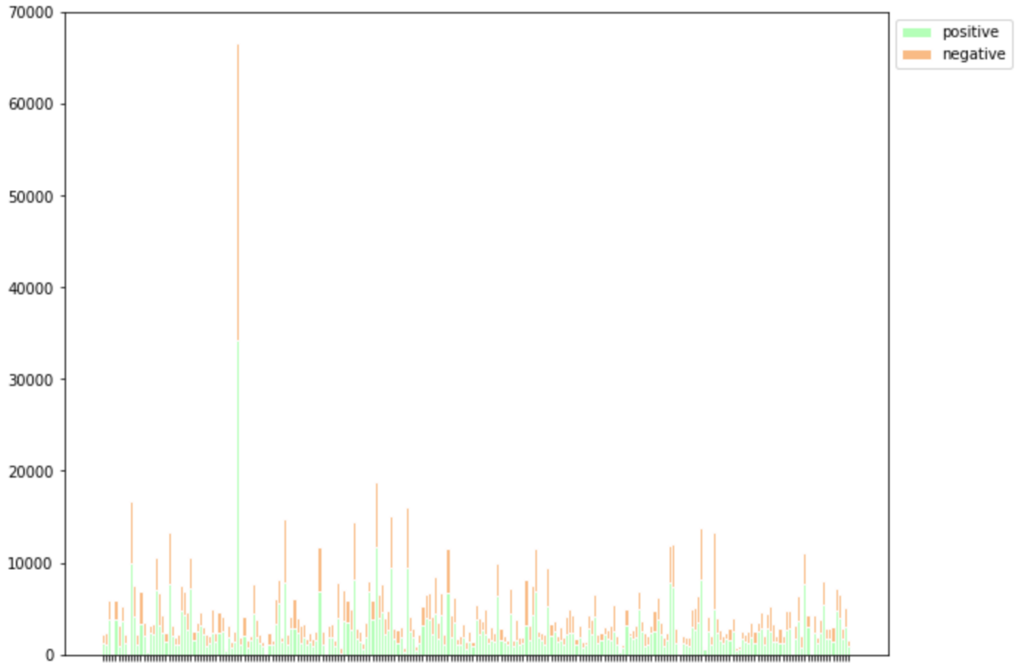


Figure 3. Distribution of positive and negative issues per project.

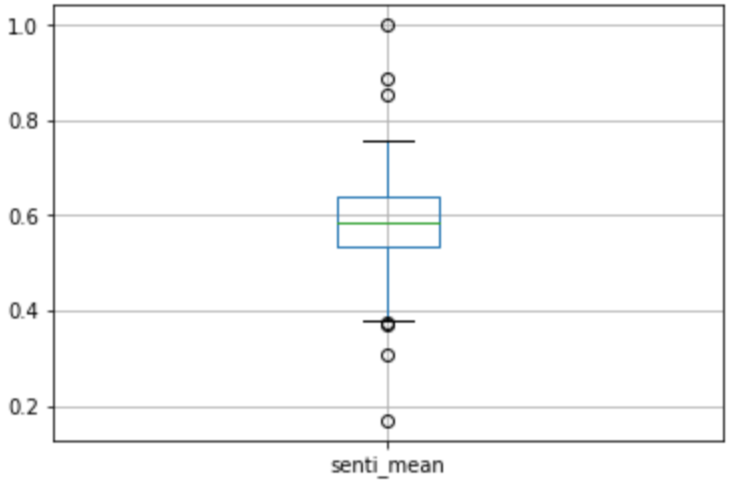
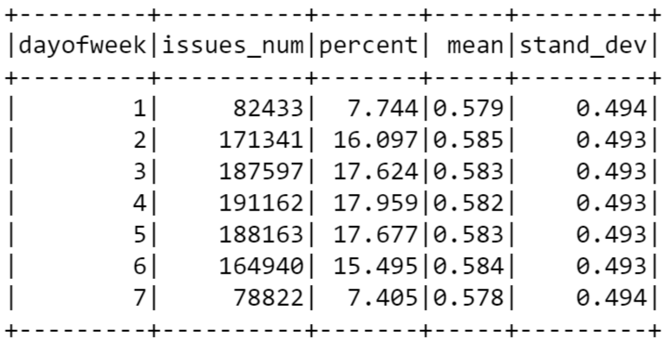


Figure 4. Box plot of all mean sentiment scores.

2) Emotions, time and date of the week

I group all of the issue comments into different categories representing the day of the week they are written. Most commit comments, 77.1%, were written during the week, while 22.9% were written during weekends. This distribution is extremely similar to the result got by Emitza Guzman et al. [3]. Besides, on Monday and Sunday, much fewer issues are produced.

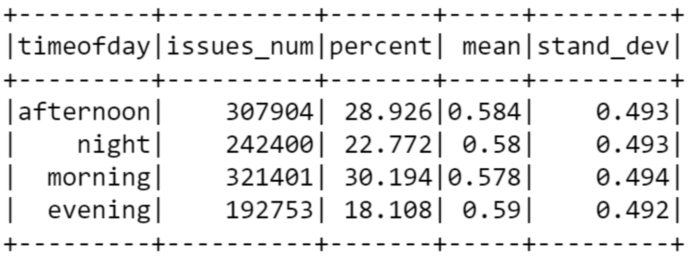


```
+---------+----------+-------+-----+---------+
|dayofweek|issues_num|percent| mean|stand_dev|
+---------+----------+-------+-----+---------+
|        1|     82433|  7.744|0.579|    0.494|
|        2|    171341| 16.097|0.585|    0.493|
|        3|    187597| 17.624|0.583|    0.493|
|        4|    191162| 17.959|0.582|    0.493|
|        5|    188163| 17.677|0.583|    0.493|
|        6|    164940| 15.495|0.584|    0.493|
|        7|     78822|  7.405|0.578|    0.494|
+---------+----------+-------+-----+---------+
```

Table 4. Emotion score average of issues grouped by weekday.

A Mann-Whitney U test of Saturday against each of the other days confirmed that issue comments are more negative on Sunday than on Tuesday, Wednesday, Thursday, Friday and Saturday, with p value at 0.001, 0.0085, 0.0426, 0.0086 and 0.0019 respectively; Issue comments are more negative on Monday than on Tuesday, Wednesday, Friday and Saturday, with p value at 0.0065, 0.0383, 0.0387 and 0.0109; Issue comments are more positive on Tuesday than on Thursday with p value at 0.0359.

Then all issue comments are separated by time of the day written during the morning [6:00-12:00), afternoon [12:00-18:00), evening [18:00-24:00) and night [0:00-6:00).



```
+---------+----------+-------+-----+---------+
|timeofday|issues_num|percent| mean|stand_dev|
+---------+----------+-------+-----+---------+
|afternoon|    307904| 28.926|0.584|    0.493|
|    night|    242400| 22.772| 0.58|    0.493|
|  morning|    321401| 30.194|0.578|    0.494|
|  evening|    192753| 18.108| 0.59|    0.492|
+---------+----------+-------+-----+---------+
```

Table 5. Emotion score average of issue comments grouped by time of the day.

It is noticeable that the percentage of issues written during the evening, at 18.108%, are apparently less than percentages of issues written during other time. I perform a Mann-Whitney U test against the issue comments of each part of the day. Issue comments are more negative in the morning than in the afternoon, evening and at night, with p value at 2.062e-6, 1.194e-17 and 0.048 respectively. Issue comments are more positive in the afternoon than at night, with p value at 0.004. Issue comments are more positive in the evening than in the afternoon and at night with p value at 5.182e-6 and 3.207e-11.

3) Emotions and time span between created and closed dates

Firstly, I filter data to find all closed issues, and group their sentiment scores by the time span between created time and closed time of issues. The medium of time spans is around 1200 days.

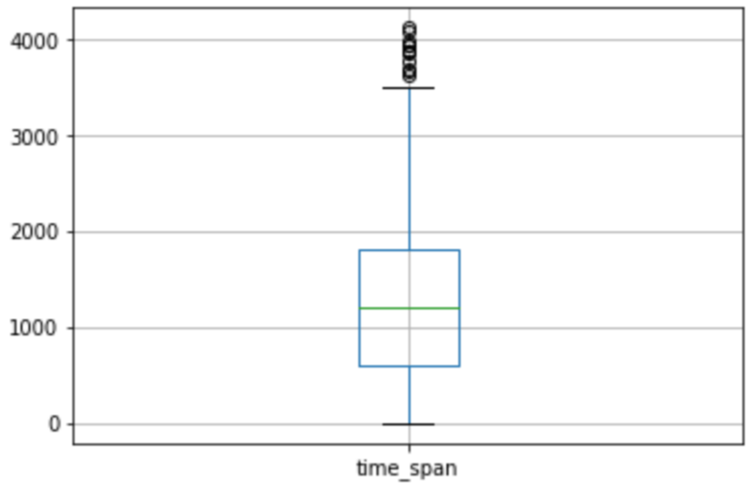


Figure 5. Boxplot of all time spans.

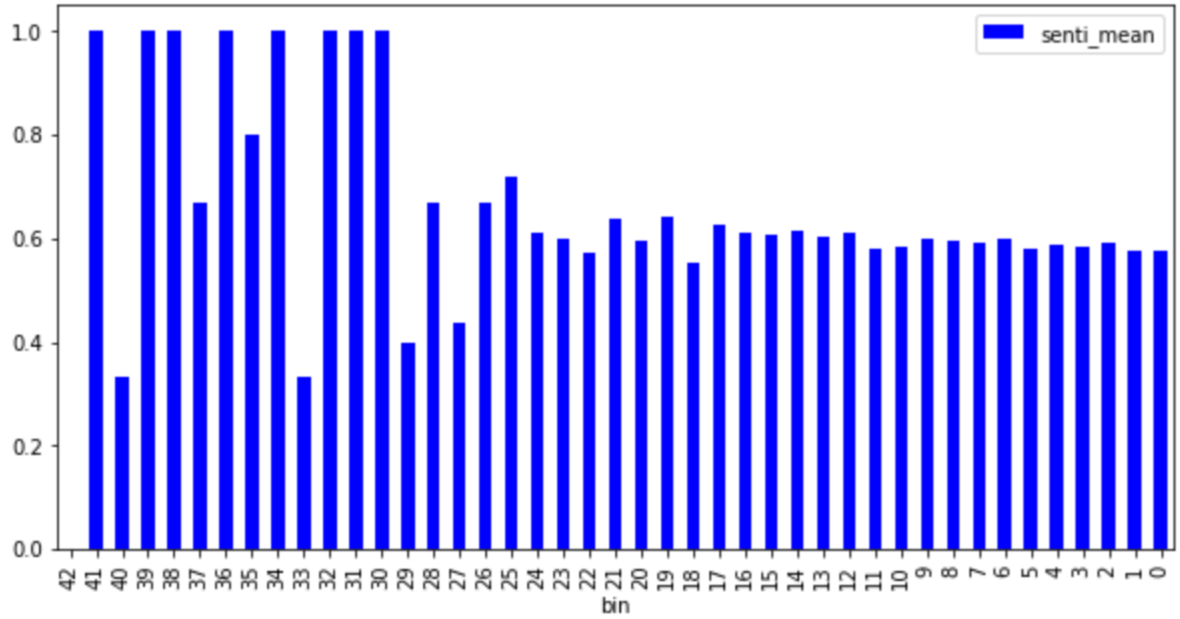


Figure 6. Emotion score average per bin, where each bin contains 100 different time spans.

Figure 6 shows the distribution of sentiment mean sorted by descending time span. It is clear that most of time spans are not influenced by sentiment in issues. And issues that tend to be more positive or negative than mean sentiment score may cause longer waiting time. I tested the Spearman correlation between the average emotion score of each time span and time spans. The correlation coefficient is 0.1096, with p value at 6.467e-8. This indicates that issues involve more emotions are more likely to cost more time before being solved than neutral issues.

5. Conclusion

In this work, the TF-IDF and logistic regression methodology is applied to calculate the sentiment scores in selected GitHub issues. It is confirmed that different project has different mean emotion scores of issues, but shares similar emotion score distribution. And by implementing the Mann-Whitney U test and the Spearman correlation to analyze sentiment scores and time characteristics, I find that emotions in issue comments are influenced by date of the week and time of the date. Meanwhile, the sentiment scores of issues have an effect on the time span between created and closed dates.

6. Future work

There are still many other factors of repositories and issues, such as the starring of project and the word counts of issues, may influence the sentiment in GitHub issues or be influenced by them, which deserves more exploration and analysis. Besides, other state of art natural language processing models for sentiment analysis are also worth comparison. It is also inspiring to examine their performance.

7. Reference

[1] Z Liao, D He, Z Chen, X Fan, Y Zhang, S Liu. Exploring the characteristics of issue-related behaviors in GitHub using visualization techniques. IEEE Access (Volume: 6), Pages: 24003 – 24015, February 2018.

[2] Murgia A, Tourani P, Adams B, OrtuM. Do developers feel emotions? An exploratory analysis of emotions in software artifacts. In: Proceedings of the working conference on mining software repositories (MSR). ACM, pp 262–271, June 2014.

[3] Emitza Guzman, David Azócar, Yang Li. Sentiment analysis of commit comments in GitHub: an empirical study. In: Proceedings of the working conference on mining software repositories (MSR). ACM, New York, MSR, pages 352–355, 2014.

[4] G. Gousios. The GHTorrent dataset and tool suite. In Proceedings of the 10th Working Conference on Mining Software Repositories, MSR, pages 233-236, 2013.

[5] E. Guzman and B. Bruegge. Towards Emotional Awareness in Software Development Teams. In Foundations of Software Engineering - FSE '13, pages 671-674, 2013.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. The Journal of Machine Learning Research, 3:993–1022, Mar. 2003.

[7] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment in short strength detection informal text. Journal of the American Society for Information Science and Technology, 61(12):2544–2558, Dec. 2010.

[8] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for

efficient text classification. arXiv preprint arXiv:1607.01759, 2016.

[9] Alessandro Murgia, Parastou Tourani , Bram Adams , Marco Ortu, An exploratory analysis of emotions in software artifacts, Proceedings of the 11th Working Conference on Mining Software Repositories, May 31-June 01, 2014, Hyderabad, India [doi>10.1145/2597073.2597086]