Sentiment Analysis over Collaborative Relationships in Open Source Software Projects

Lingjia Li[‡], Jian Cao*[‡] and David Lo[§]
[‡]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
[§]School of Information Systems, Singapore Management University, 178902, Singapore
Email: [‡]{jessie_llj,cao-jian}@sjtu.edu.cn, [§]davidlo@smu.edu.sg

***Abstract*—Sentiments and collaboration efficiency are key factors in the success of the open source software (OSS) development process. However, in the software engineering domain, no studies have been conducted to analyze the effect between collaborators' sentiments, and the role of sentiment in collaborative relationships during the development process. In this study, we apply sentiment analysis and statistical analysis on collaboration artifacts over five projects on GitHub. We use sentiment consistency to quantify the relation between sentiments in collaborative relationships. It is found that sentiment consistency is positively correlated with the closeness of collaborative relationships and collaborators' overall sentiment states. We also perform the Granger causality test and network analysis to study the impact of sentiment consistency on a time series basis. It is found that positive consistent sentiments not only improve collaboration willingness to the utmost extent, followed by inconsistent and negative consistent sentiments, they also boost the closeness of the entire project community. These findings can be applied to develop better OSS project monitoring tools and improve project management by taking developers' sentiments during collaborations into consideration.**

***Index Terms*—Sentiment analysis, Human factors, GitHub, Collaborative and social computing, Project management**

I. Introduction

Software development is a highly collaborative activity where developers work on collaborative tasks and interact with shared artifacts to create and maintain a complex software system. The efficiency of collaboration is a distinguishing factor in the success or failure of many modest to large software development organizations [1]. Therefore, how to improve collaboration efficiency is an important issue in project management. Many factors affect collaboration efficiency, including ease of use of technology, trust between the teams and well-defined task structure [2], etc.

Basic emotions include *anger*, *disgust*, *fear*, *joy*, *sadness*, and *surprise*. Emotions can generally imply people's sentiments which are usually classified into *positive*, *negative* and *neutral*. Such sentiments in professional work can affect creativity, group rapport, user focus, and job satisfaction [3]. In software development, happy developers have higher debugging performance [4], self-assessed productivity and solve problems better [5]. Studies that perform sentiment analysis on software artifacts find that positive sentiments in development activities increase the number of commit files [6] and decrease issue resolution time [7]. These findings highlight the role of sentiment in software development and suggest that understanding its key factors can help improve developers' performmance.

Sharing feedback in the form of sentiments can positively affect online trust in inter-user collaborations among Wikipedia editors [8]. In the education domain, it is also found that emotions have an impact on forming successful collaborative relationships [9]. However, in the software engineering (SE) domain, there is no study to analyze the effect between collaborators' sentiments, and the role of sentiment in collaborative relationships. Therefore, in this study we focus on the sentiment over collaborative relationships during software development. Specifically, we are interested in the collaborative relationships in open source software (OSS) projects, where the work is volunteer-driven, hence developers' enjoyment plays a dominant role during the developments [10]. To identify the key factors behind collaborators' sentiment relations and understand how these relations interact with collaborative relationships is important for managing OSS project. On one hand, it can help develop better tools for monitoring sentiments to resolve potential risks. On the other hand, effective strategies can be adopted to coordinate the development process, for example, recommending compatible developers by taking their sentiment effects during collaboration into consideration.

We perform sentiment analysis on issue comments in GitHub and define sentiment consistency to measure the relationship between collaborators expressed by their sentiments. We intend to answer the following research questions:

RQ1: Are collaborators' sentiments more consistent than those between other developers?

We compare sentiment consistency over collaborative and non-collaborative relationships to find out whether developers' sentiments are affected by the collaborative relationship.

RQ2: What factors does sentiment consistency correlate with in a collaborative relationship?

We examine closeness of collaborative relationships, collaborators' overall sentiment states and position difference to understand how collaborative relationships affect sentiment consistency. From an organizational standpoint, this can provide guidelines to promote effective collaborative relationships.

RQ3: What is the impact of sentiment consistency on the formation of collaborative relationships?

DOI reference number: 10.18293/SEKE2020-030
*Corresponding Author

We investigate whether sentiment consistency has an inverse impact on collaborations. This effort aims to help managers coordinate collaborators' sentiments targetedly.

II. Related Work

A. Sentiment Analysis for OSS projects

Sentiment analysis [11] uses natural language processing, text analysis and computational techniques to automate the extraction or classification of sentiments from texts. There are a number of mature and publicly available tools like SentiStrength [12], Stanford NLP sentiment analyser [13] and Natural Language Text Processing (NLTK) [14]. Applying sentiment analysis to SE communities is a relative new research field. However, sentiment analysis tools trained or evaluated on non-technical datasets can generate unreliable results on SE datasets [15]. Therefore, some tools have been developed specifically for SE domain, like SentiStrength-SE [16], Senti4SD [17] and SentiCR [18].

Based on these tools, researchers are focusing their efforts on understanding how different factors interact with developers' sentiments. Some studies explore the factors that affect developers' sentiments. Pletea, Vasilescu, and Serebrenik [19] analyze commits and pull requests on GitHub and finds that more negative emotions are expressed in security-related discussions. Java projects are found to attract more negative comments while projects with more distributed teams attract more positive comments [20]. A study investigating commit logs on GitHub finds that Tuesday's comments have the most negative sentiments [6].

Other studies evaluate how developers' sentiments impact their performance to reveal what kinds of sentiments benefit the development process. An online survey [21] shows that *anger* can enhance developers' productivity, while *frustration* and *disgust* may bring risks. A study on the OSS project GENTOO shows that developers expressing strong emotions in issue trackers are more likely to become inactive in the projects they contribute to [22]. Ortu et al. [7] build a logistic regression model on 560k JIRA comments and find that the more positive the average sentiment, the faster an issue is fixed.

B. Factors influencing successful collaborative relationships

The success of software development largely depends on developers' collaboration efficiency and many factors influence the formation of successful collaborative relationships. The work by Kotlarsky and Oshri [23] suggests that human-related issues, such as rapport and transactive memory, are important for collaborative work. Joint intention, sharing of goals, plans and knowledge of the environment, awareness of the roles and responsibilities and team awareness are identified as the capabilities needed by an effective team [24]. Trust is another factor in forming successful collaborative relationships [25]. Perrault et al. [26] prove having learning as a purpose and sharing leadership to be success factors. Unfortunately, the role of sentiment in collaborative relationships has not been investigated yet.

TABLE I
Details of Projects

Project	Language	#Issues	#Developers	Avg. Iss. per Dev.
Three.js	JavaScript	5465	2011	2.72
Pandas	Python	22854	5713	4.0
IPython	Python	10172	3413	2.98
gRPC	C++	14828	3142	4.72
OpenRA	C#	6026	682	8.84

III. Proposed methods

In this section, we describe our dataset and data processing methods for the subsequent analyses[1].

A. Dataset

GitHub is a popular code repository site for many well-known and active OSS projects. In GitHub, each project has its own repository and the history of the source code, commits, issues, and other related data are all publicly accessible. We obtain the dataset through GitHub REST API[2]. TABLE I lists the five mid-to-large scale projects we focus on. To ensure a high sample coverage [27] of the sampled data, the selection of programming languages is basically in line with the Top Languages[3] on GitHub. Besides, we take average number of issues each developer participates into account.

B. Data Extraction

1) Identification of Collaborative Relationships: Generally, collaborative relationships are formed when two people work together to accomplish common goals. In GitHub, issue reports are used by team members to ask for advice, and express and share opinions related to software maintenance and evolution [28]. In our study, collaboration between two developers is defined as the issue resolution process they both participate in. A collaborative relationship is identified when two developers both post comments under an issue.

2) Sentiment Analysis and Sentiment Consistency: Sentiments are commonly expressed in developers' issue comments [29]. We perform sentiment analysis using SentiCR and retrain the classifier by a gold standard [30] containing 3,000 manually labeled issue comments of ten OSS projects on GitHub.

To quantify the relation between sentiments expressed by collaborators in the software development process, we define sentiment consistency, which is identified through comparing collaborators' sentiment polarities. Two comments with the same sentiment polarity (both positive/negative/neutral) are considered to be sentiment consistent. Two comments with opposing sentiment polarities (one detected as positive while the other detected as negative) are considered to be sentiment inconsistent.

For a collaborative relationship involving two developers d_j and d_k, sentiment consistency in issue ι, denoted as $C_{\langle d_j, d_k \rangle}(\iota)$, is the number of sentiment-consistent comment pairs they post

[1]Code and data are released on http://doi.org/10.5281/zenodo.3608892
[2]https://developer.github.com/v3/
[3]https://octoverse.github.com/

in ι divided by the total number of comment pairs they post in ι. Formally,

$$C_{\langle d_j,d_k\rangle}(\iota) = \frac{\sum\limits_{p_i \in P} c_{d_j}^{p_i}(\iota) c_{d_k}^{p_i}(\iota)}{c_{d_j}(\iota) c_{d_k}(\iota)} \tag{1}$$

where $P = \{positive,\ negative,\ neutral\}$. $c_{d_j}^{p_i}(\iota)$ and $c_{d_k}^{p_i}(\iota)$ represent the number of comments posted by d_j and d_k in issue ι with polarity p_i respectively. $c_{d_j}(\iota)$ and $c_{d_k}(\iota)$ represent the total number of comments in issue ι by d_j and d_k. Sentiment consistency over collaborative relationship $\langle d_j, d_k\rangle$ is then formulated as the mean of sentiment consistency in all the n issues co-participated by d_j and d_k:

$$C_{\langle d_j,d_k\rangle} = \frac{\sum\limits_{i=1}^{n} C_{\langle d_j,d_k\rangle}(\iota_i)}{n} \tag{2}$$

C. Dynamic Collaboration Sentiment Network

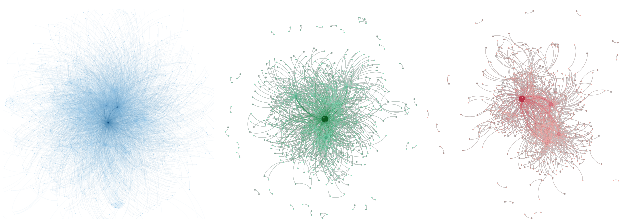


Fig. 1. From left to right: Original, Positive and Negative Network of Three.js.

1) Network construction: A (static) collaboration network N^t is defined as a network of collaborative relationships in which each node is a developer, and two nodes are connected if there is a collaborative relationship between these two developers during period t. Each edge is associated with a weight corresponding to the times of collaborations.

Besides the original collaboration network, we construct a positive-consistent and a negative-consistent sentiment collaboration network from the extracted data. This is achieved through consistency filtering: we only keep the sentiment-consistent collaborations and remove the others. In the network view, this means that the weight of an edge is reduced by the times of collaborations in which the two collaborators (nodes) do not share common positive or negative sentiments. Figure 1 shows the three obtained networks of Three.js. These networks provide useful informaiton of collaborative relationships as well as corresponding sentiment effects inside the project, so that we can interpret how sentiment consistency impacts collaborative relationships from a structural point of view.

Moreover, we construct dynamic networks to analyze the evolution of each network structure. A dynamic collaboration network N is a sequence of collaboration networks corresponding to different periods of time.

$$N := (N^{t_1}, N^{t_2}, ..., N^{t_n}) \tag{3}$$

where the periods $t_1, ..., t_n$ are obtained by dividing the overall development time into half-year intervals.

2) Network Analysis: We want to analyze the collaboration networks in terms of connectivity, community structure and betweenness to identify positive and negative sentiments' different impacts on collaborative relationships. We focus on three global measures: mean clustering coefficient, modularity and average betweenness centrality to mitigate the influence of network size.

TABLE II
SENTIMENT CONSISTENCY BETWEEN COLLABORATORS AND NON-COLLABORATORS

	Collaborators mean	Collaborators std	Non-collaborators mean	Non-collaborators std	p for t-test
Three.js	0.471	0.176	0.4464	0.148	<0.0001
Pandas	0.517	0.189	0.483	0.161	<0.0001
IPython	0.475	0.162	0.448	0.139	<0.0001
gRPC	0.581	0.211	0.545	0.174	<0.0001
OpenRA	0.635	0.172	0.543	0.162	<0.0001

a) Mean clustering coefficient: The clustering coefficient [31] of a node is defined as:

$$c_i = \frac{2n_i}{k_i(k_i-1)} \tag{4}$$

where n_i denotes the number of edges between the k_i neighbors of node i. The intuition is that $k_i(k_i-1)/2$ edges can exist between k_i nodes, and the clustering coefficient reflects the fraction of existing edges between neighbors divided by the total number of possible edges. We employ the mean clustering coefficient to measure to what degree collaborators tend to cluster together in different networks.

b) Modularity: Modularity is the standard measure to quantify the strength of a community structure [32]. Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules. In our context, modularity indicates whether collaborators with consistent sentiment are divided into separate groups or integrated into a cohesive whole.

c) Average Betweenness Centrality: For a node v, betweenness centrality [33] is the sum of the fraction of all-pairs shortest paths that pass through v in the network:

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \tag{5}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest (s,t)-paths and $\sigma(s,t|v)$ is the number of those paths that contain node v in between. High betweenness centrality indicates that the person plays the role of gatekeeper in the social network, with the potential to disrupt connections between various end points.

IV. RESULTS AND FINDINGS

A. Collaborators vs. Non-collaborators

Sentiment consistency over non-collaborative relationships is the sentiment-consistent comment pairs divided by the total comment pairs of two developers in issues excluding the n co-participated ones. Formally,

$$C'_{\langle d_j,d_k\rangle}(\iota) = \frac{\sum\limits_{p_i \in P} c_{d_j}^{p_i} c_{d_k}^{p_i} - \sum\limits_{i=1}^{n} \sum\limits_{p_i \in P} c_{d_j}^{p_i}(\iota_i) c_{d_k}^{p_i}(\iota_i)}{c_{d_j} c_{d_k} - \sum\limits_{i=1}^{n} c_{d_j}(\iota_i) c_{d_k}(\iota_i)} \tag{6}$$

TABLE III
SPEARMAN COEFFICIENTS BETWEEN THREE FACTORS AND SENTIMENT CONSISTENCY

	#Common Issues		Sentiment		Position Difference	
Three.js	0.116	***	0.357	***	0.057	*
Pandas	0.062	**	0.328	***	0.001	0.942
IPython	0.122	***	0.273	***	-0.035	0.132
gRPC	0.27	***	0.107	***	-0.227	***
OpenRA	0.233	***	0.44	***	-0.076	0.107

*$p < 0.05$, **$p < 0.01$, ***$p < 0.001$

TABLE IV
SPEARMAN COEFFICIENTS BETWEEN COLLABORATORS' POSITION DIFFERENCE AND TIMES OF COLLABORATIONS

Repository	**Coefficient**	**p-value**
Three.js	-0.085	<0.001
Pandas	-0.356	<0.0001
IPython	-0.304	<0.0001
gRPC	-0.629	<0.0001
OpenRA	-0.423	<0.0001

The means of sentiment consistency for collaborators and non-collaborators are compared through independent t-tests. It is estimated from the distribution plots that our data are normally distributed. It can be found in TABLE II that collaborators share more consistent sentiments than non-collaborators.

B. Factors Influencing Sentiment Consistency

We investigate the correlation of three factors with sentiment consistency over collaborative relationships, i.e., closeness of the collaborative relationship, the collaborators' overall sentiment state and the position difference between them. The model outputs are listed in TABLE III.

1) Closeness of Collaborative Relationship: We measure the closeness of a collaborative relationship through the number of issues each pair of collaborators co-participate in. Its Spearman correlation coefficient with collaborators' sentiment consistency is calculated. It is found that sentiment consistency is higher in collaborators with more co-participated issues.

2) Collaborators' Overall Sentiment State: A study on Twitter shows that the positive sentiment is contagious because community members increasingly share positive tweets more than negative ones over time [34]. We try to validate whether this effect can be applied to collaborations in OSS development. A developer's sentiment state is measured through the number of non-negative comments divided by the total number of his/her comments. We analyze the correlation between the average sentiment state of two collaborators and their sentiment consistency over the collaborative relationship. A positive correlation is found.

3) Position difference between collaborators: In this study, we want to identify whether different positions between collaborators can impact sentiment consistency in the OSS development process. We employ node degree of the basic collaboration network in Section 3.3.1 to represent a developer's position in the project. The normalized position difference of two collaborators is formulated as $\frac{|d_1-d_2|}{max(d_1,d_2)}$, where d_1 and d_2 stand for the node degree. Its Spearman correlation coefficient with sentiment consistency is then measured.

A negative correlation between the sentiment consistency and collaborators' position difference is found in gRPC and OpenRA. We further investigate the correlations between collaborators' position difference and times of collaborations. The results are shown in TABLE IV.

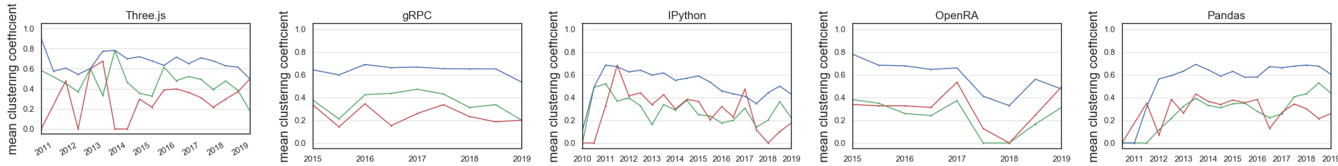
It can be found that there are negative correlations between collaborators' position difference and times of collaborations in gRPC and OpenRA. Weak negative correlations appear in Pandas and IPython while there is no correlation in Three.js. Furthermore, by comparing the attributes of these five projects, it can be found that the correlation between collaborators' position difference and times of collaborations is affected by the average number of issues that a developer participated in (See TABLE I). In projects that developers participate in quite a few issues (gRPC and OpenRA), the developers of similar positions tend to have more collaborations, which contribute to higher sentiment consistency. On the contrary, in projects that developers only participate in a small number of issues (Three.js), positions do not impact their collaborations. Actually, there are no big differences between developers' positions in these projects.

C. Impacts of Sentiment Consistency on Collaboration

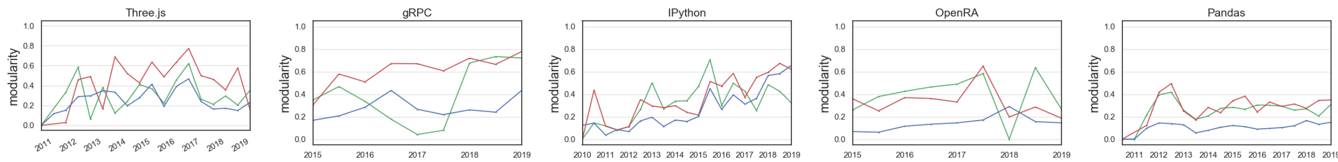
We employ the Granger causality test [35] to determine whether sentiment consistency has a causal relationship with collaboration willingness. For two time series X and Y, if Y can be better predicted using the lagged values of both X and Y than using the lagged values of Y alone, then X is said to Granger cause Y. In this context, we investigate whether the occurrence of collaborations is Granger caused by the increase of sentiment consistency in a prior period.

$$\begin{aligned} Y_t &= \mu + \sum_{i=1}^{L} \alpha_i Y_{t-i} + \sum_{i=1}^{L} \beta_i X_{t-i} \\ L &= max.\ no.\ of\ lags \end{aligned} \tag{7}$$

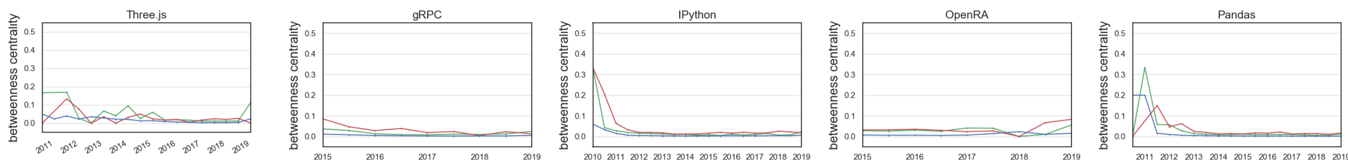
For each pair of collaborators, we extract the frequency of sentiment consistency and the frequency of collaborations within a week. We run the adfuller test to select stationary time series, which is the precondition required by Granger causality tests. Then we run an independent Granger causality test on the two time series. The maximum time lags are 2, 4, 8 and 12 weeks respectively. The numbers of significant causal relationships are compared in TABLE V. It can be found that the effect of sentiment consistency on collaborations is more significant than sentiment inconsistency; the effect of positive consistency is more significant than negative consistency.
We further investigate the evolution of network measures of the three constructed networks. Figure 2 illustrates the results. As can be seen, the mean clustering coefficients of negative collaboration networks are lower than the positive ones, implying that collaborators with positive-consistent sentiments are more clustered and interconnected. The modularity of negative collaboration networks is higher than positive ones, implying that developers with negative-consistent sentiments



(a) Mean clustering coefficient



(b) Modularity



(c) Average betweenness centrality

Fig. 2. Evolution of Network Measures in Original *(blue)*, Positive *(green)* and Negative *(red)* Collaboration Networks in Five Projects.

TABLE V
TOTAL NUMBER OF COLLABORATIVE RELATIONSHIPS WITH SIGNIFICANT CAUSAL EFFECTS OVER DIFFERENT SENTIMENT POLARITIES.

Polarity	**#Collaborations**	**Total**	***p*-value**
consistent	1782 (47.9%)	3807	3.80e-35
inconsistent	1109 (27.5%)	4029	
positive consistent	1120 (28.2%)	3973	7.22e-71
negative consistent	686 (16.7%)	4100	

are more densely distributed inside small groups. The betweenness centrality coefficients are generally higher in negative collaboration networks, implying that negative developers play a more important role in sentiment propagation through collaborations than positive developers do.

V. IMPLICATIONS

Although how different factors interact with developers' sentiments during development has been studied, there is no study to analyze the role of sentiment in collaborative relationships. Our findings indicate that positive sentiment linkage can boost collaborators' closeness and should be encouraged in software development to foster a better collaboration ecology.

It is also suggested that negative sentiment effects are more likely to be reduced than augmenting positive ones through adjustments and the reassignment of collaborators based on the network features as well as factors influencing sentiment consistency, so as to maximize collaboration willingness and efficiency. The results also indicate that developers of different positions in the collaboration network tend to have fewer collaborations in practice, which may reduce sentiment consistency accordingly. This inspires us that we should promote the collaborations among developers of different positions in the collaboration network. More specifically, to encourage positive developers of high degree (i.e., of central position) to cooperate with negative developers of low degree (i.e., of peripheral position) can bring more gains to projects.

These findings tell us when we are going to coordinate the OSS projects, we should take the consistency of developers' sentiments into account in order to promote their collaborations in a task. As a measure of implementations, the consistency of developer' sentiments can be monitored so that we can take appropriate measures to regulate the organization of the development process. It also encourages us to incorporate these new features into the future monitoring tools.

VI. THREATS TO VALIDITY

In this preliminary study, we only study a sample of projects. We take into account different developer densities and programming languages to ensure the diversity of samples and the generalizability of our results.

We use SentiCR, a customized sentiment analysis tool for SE domain for sentiment analysis, and retrain it with a gold standard on GitHub issues. However, misclassifications may still exist and bring noise to our dataset.

Another threat is the definition of collaborative relationships. Collaborations on the same file are not taken into account because it is difficult to extract the sentiments during this kind of collaboration. Additionally, all the comments in an issue are considered in the calculation of sentiment consistency between two developers because it is difficult to determine whether they are in the same thread of conversation. However, from our observation, two developers may discuss irrelevant tasks in one issue, leading to a mis-detected collaborative relationship. Topic extraction or NLP techniques can be further adopted to address this issue.

VII. Conclusions

We use sentiment consistency to quantify the relation between sentiments in collaborative relationships. Our results show that collaborators share more consistent sentiments and sentiment consistency has a positive correlation with the closeness of collaborative relationships and collaborators' overall sentiment states. It has a negative correlation with position difference in projects that developers participate in quite a few issues. Results of the Granger causality test show that positive consistent sentiments have the most significant impact on collaboration willingness, followed by inconsistent and negative-consistent sentiments. Network analysis shows negative consistent collaborators are more alienated and distributed in small groups, while positive consistent collaborators tend to cluster into a cohesive whole. In a follow-up study, we plan to include more projects and refine our methods for collaborative relationship detection. We also plan to research the main causes of sentiment fluctuation over collaborative relationships, and develop specific strategies for monitoring such sentiment phenomena. We hope our results can motivate further research to help coordinate developers' collaborations and provide better tools for higher serenity and productivity in software development communities.

Acknowledgement

This work is partially supported by National Key Research and Development Plan(No.2018YFB1003800).

References

[1] Grady Booch and Alan W Brown. "Collaborative development environments". In: *Advances in computers* 59.1 (2003), pp. 1–27.

[2] H Keith Edwards and Varadharajan Sridhar. "Analysis of the effectiveness of global virtual teams in software engineering projects". In: *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*. IEEE. 2003, 9–pp.

[3] Cynthia D Fisher and Neal M Ashkanasy. "The emerging role of emotions in work life: An introduction". In: *Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior* 21.2 (2000), pp. 123–129.

[4] Iftikhar Ahmed Khan, Willem-Paul Brinkman, and Robert M Hierons. "Do moods affect programmers' debug performance?" In: *Cognition, Technology & Work* 13.4 (2011), pp. 245–258.

[5] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. "Happy software developers solve problems better: psychological measurements in empirical software engineering". In: *PeerJ* 2 (2014), e289.

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

[7] Marco Ortu et al. "Are bullies more productive?: empirical study of affectiveness vs. issue fixing time". In: *Proceedings of the 12th Working Conference on Mining Software Repositories*. IEEE Press. 2015, pp. 303–313.

[8] Mihai Grigore and Christoph Rosenkranz. "Increasing the Willingness to Collaborate Online: an Analysis of Sentiment-Driven Interactions in Peer Content Production." In: vol. 4. Jan. 2011.

[9] Jane Conoley and Collie Conoley. "Why Does Collaboration Work? Linking Positive Psychology and Collaboration". In: *Journal of Educational and Psychological Consultation - J EDUC PSYCHOLOGICAL CONS* 20 (Feb. 2010), pp. 75–82.

[10] Karim R Lakhani and Eric Von Hippel. "How open source software works:"free" user-to-user assistance". In: *Produktentwicklung mit virtuellen Communities*. Springer, 2004, pp. 303–339.

[11] Bing Liu et al. "Sentiment Analysis and Subjectivity." In: *Handbook of natural language processing* 2.2010 (2010), pp. 627–666.

[12] Mike Thelwall et al. "Sentiment strength detection in short informal text". In: *Journal of the American Society for Information Science and Technology* 61.12 (2010), pp. 2544–2558.

[13] Richard Socher et al. "Recursive deep models for semantic compositionality over a sentiment treebank". In: *Proceedings of the 2013 Conference on Empirical mMthods in Natural Language Processing*. 2013, pp. 1631–1642.

[14] Edward Loper and Steven Bird. "NLTK: the natural language toolkit". In: *arXiv preprint cs/0205028* (2002).

[15] Robbert Jongeling et al. "On negative results when using sentiment analysis tools for software engineering research". In: *Empirical Software Engineering* 22.5 (2017), pp. 2543–2584.

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

[17] Fabio Calefato et al. "Sentiment polarity detection for software development". In: *Empirical Software Engineering* 23.3 (2018), pp. 1352–1382.

[18] Toufique Ahmed et al. "SentiCR: a customized sentiment analysis tool for code review interactions". In: *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*. IEEE Press. 2017, pp. 106–111.

[19] Daniel Pletea, Bogdan Vasilescu, and Alexander Serebrenik. "Security and emotion: sentiment analysis of security discussions on GitHub". In: *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM. 2014, pp. 348–351.

[20] Emitza Guzman, David Azócar, and Yang Li. "Sentiment analysis of commit comments in GitHub: An empirical study". In: *11th Working Conference on Mining Software Repositories, MSR 2014 - Proceedings* (May 2014). DOI: 10.1145/2597073.2597118.

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

[22] David Garcia, Marcelo Serrano Zanetti, and Frank Schweitzer. "The role of emotions in contributors activity: A case study on the Gentoo community". In: *2013 International Conference on Cloud and Green Computing*. IEEE. 2013, pp. 410–417.

[23] Julia Kotlarsky and Ilan Oshri. "Social ties, knowledge sharing and successful collaboration in globally distributed system development projects". In: *European Journal of Information Systems* 14.1 (2005), pp. 37–48.

[24] Katia Sycara and Gita Sukthankar. "Literature review of teamwork models". In: *Robotics Institute, Carnegie Mellon University* 31 (2006), p. 31.

[25] Valerie Lynne Herzog. "2000 International Student Paper Award Winner: Trust Building on Corporate Collaborative Project Teams". In: *Project Management Journal* 32.1 (2001), pp. 28–37.

[26] Ellen Perrault et al. "Working together in collaborations: Successful process factors for community collaboration". In: *Administration in social work* 35.3 (2011), pp. 282–298.

[27] Meiyappan Nagappan, Thomas Zimmermann, and Christian Bird. "Diversity in software engineering research". In: *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. ACM. 2013, pp. 466–476.

[28] Alessandro Murgia et al. "Do developers feel emotions? an exploratory analysis of emotions in software artifacts". In: *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM. 2014, pp. 262–271.

[29] Francisco Jurado and Pilar Rodriguez. "Sentiment Analysis in monitoring software development processes: An exploratory case study on GitHub's project issues". In: *Journal of Systems and Software* 104 (2015), pp. 82–89.

[30] Jin Ding et al. "Entity-level sentiment analysis of issue comments". In: *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*. ACM. 2018, pp. 7–13.

[31] Stefano Boccaletti et al. "Complex networks: Structure and dynamics". In: *Physics reports* 424.4-5 (2006), pp. 175–308.

[32] Mark EJ Newman. "Modularity and community structure in networks". In: *Proceedings of the National Academy of Sciences* 103.23 (2006), pp. 8577–8582.

[33] Linton C Freeman. "A set of measures of centrality based on betweenness". In: *Sociometry* (1977), pp. 35–41.

[34] Noha Alduaiji and Amitava Datta. "An Empirical Study on Sentiments in Twitter Communities". In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2018, pp. 1166–1172.

[35] Clive WJ Granger. "Investigating causal relations by econometric models and cross-spectral methods". In: *Econometrica: Journal of the Econometric Society* (1969), pp. 424–438.