How Developers and Tools Categorize Sentiment in Stack Overflow Questions - A Pilot Study

Niloofar Mansoor
University of Nebraska - Lincoln
Lincoln, Nebraska USA 68588
niloofar@huskers.unl.edu

Cole S. Peterson
University of Nebraska - Lincoln
Lincoln, Nebraska USA 68588
Cole.Scott.Peterson@huskers.unl.edu

Bonita Sharif
University of Nebraska - Lincoln
Lincoln, Nebraska USA 68588
bsharif@unl.edu

***Abstract*—The paper presents results from a pilot questionnaire-based study on ten Stack Overflow (SO) questions. Eleven developers were tasked with determining if the SO question sentiment was positive, negative or neutral. The results from the questionnaire indicate that developers mostly rated the sentiment of SO questions as neutral, stating that they received little or no emotional feedback from the questions. Tools that were designed to analyze Software Engineering related texts (SentiStrength-SE, SentiCR, and Senti4SD) were on average more closely aligned with developer ratings for a majority of the questions than general purpose tools for detecting SO question sentiment. We discuss cases where tools and developer sentiment differ along with implications of the results. Overall, the sentiment tool output on the question title and body is more aligned with the developer rating than just the title alone. Since SO is a very common medium of technical exchange, we also report that adding code snippets, short titles, and multiple tags were top three features developers prefer in SO questions in order for it to be answered quickly.**

***Index Terms*—sentiment analysis, Stack Overflow, empirical study, sentiment tools, software engineering**

I. Introduction

Stack Overflow (SO) is a popular medium for programmers and software engineers to cooperate with others, ask questions, and seek answers to design and/or coding problems they face during development. This community-based question and answer site has risen in popularity in the last decade and programmers of varying levels use it to their benefit and to share knowledge with others. Similar to any other social platform, the success of any content is affected by different factors. In SO, the questions are deemed successful if there are numerous, accurate, and timely answers to it. Formulating good quality questions increases the chance of receiving answers and support. Thus, it is important to know what factors are important in writing good questions. Good questions will also enhance crowdsourced knowledge and will be beneficial for people who want to find specific expert answers to questions. Calefato el al. [1] defined some features and empirically determined whether their presence would contribute to the success of a SO question. In addition to those features, they also assessed how the emotional valence (negative, neutral, and positive sentiment) of a question affects its success. They realized that the valence is an important factor and neutral questions have a relatively higher chance of being successful. This finding, and the fact that sentiment and emotional awareness is a growing sub-field in software engineering [2] evidenced by different tools that can be used to assess the sentiment of natural and technical language text, we decided to analyze both the features and sentiment of SO questions in a pilot study with eleven developers, and compare the results with state-of-the-art sentiment analysis tools as well to see how they work on SO questions.

Prior work in the field has warned about strong limitations of current state of the art tools [3] applied to software engineering datasets. Novielli et al. [4] reported on a benchmark study for assessing the performance of three different domain-specific sentiment analysis tools that are built for performing sentiment analysis for Software Engineering related texts (Senti4SD [5], SentiStrengthSE [6], SentiCR [7]). Imtiaz et al. [8] utilizes sentiment analysis tools on GitHub comments to explore the sentiment in developer discussions, and they find that the tools have low agreement with human ratings. Lin et al. [9] investigate the accuracy of commonly used tools in sentiment analysis of SE text as well.

In this short paper, we further seek to validate (via a pilot study), how SO questions fair with sentiment analysis tools as well as how they compare when humans (developers) rate them for sentiment. Instead of focusing primarily on generic sentiment analysis tools such as SentiStrength, we compare how human annotation fairs with software engineering specific tools such as SentiCR (2017), Senti4SD (2018) and SentiStrength-SE (2018). To provide a baseline of comparison to generic sentiment tools, we use NLTK and the Stanford NLP tools. Human annotations are then compared with both software engineering tools as well as generic tools.

The two research questions we address are:

- RQ1: How do developers rate SO questions in terms of sentiment?
- RQ2: How do state-of-the-art sentiment analysis tools' output match with developer rating of sentiment on SO questions?

The research questions are focused in determining if there is a match between how a developer rates a SO question's sentiment compared to a tool. This partially seeks to validate prior results [10] but uses three additional software engineering specific tools (Senti-CR, SentiStrength-SE, and Senti4SD). The main reason we focus on SO questions is because this is the first thing people read on Stack Overflow, and programmers decide whether or not to answer primarily based on the SO question text.

II. Pilot Study Design

We selected ten SO posts for participants to read and rate the sentiment of. These were selected to ensure that the post attributes were evenly distributed amongst the number of answers the question received, number of votes, and poster's reputation score. The titles of the questions range from 6 to 12 words and the bodies of the questions range from 22 to 100 words excluding code snippets. Additional information about the questions selected for this study are shown in Table I.

Ten developers participated in this study. All of the participants were male undergraduate or graduate Computer Science students. Two students were between 18-22 years old, four of them were 23-27 years old, and five of them were over 27 years old. Participants were asked to read the SO question. Once they finished reading, they were asked to do a feature rating and a sentiment rating. This process continued for each of the ten questions they were asked to rate. They were also optionally allowed to comment on their answer via a text box. The replication package [11] provides all study materials.

Feature Rating: With regards to the feature rating of the SO questions, participants were asked what features they would prefer to see in the question such that the likelihood of it getting answers would be increased. The participants had six choices and could select any number of the features that were presented to them. These features are as follows: Use of code snippets, Use of multiple tags, Multiple URLs, Short title and body length, Low usage of uppercase characters, and Other which allowed the participant to specify an additional feature. These features presented to developers were taken from the work of Calefato el al. [1]. We decided to add the feature rating to complement results from the sentiment rating and to revalidate the results of Calefato et al. [1] on a different dataset. The main difference in our questionnaire is that we asked about the feature rating on specific questions, whereas Calefato et al. asked developers what features they prefer on SO questions in general.

Sentiment Rating: For each SO question, the participants rated the sentiment of the question post. They were asked to rate each question as having more positive sentiment, more negative sentiment, or neutral sentiment, and they were required to explain their choice. The participants were not given any guidelines on what constitutes a positive, neutral, or negative sentiment and it was up to them to interpret the sentiment.

III. Data Pre-processing

Before analyzing the data, we collected and aggregated the forms the participants filled out into a single master file. We looked at the reasons the participants gave for the sentiment they assigned the question, and for some participants, they gave an explanation that was based on the answers such as ”The number of answers here was high”. Since we only focus on question sentiment, the responses that mentioned answers were filtered and not used when calculating user sentiment rating of the question but were used in the feature rating aggregation. The title text and the combination of title text and body were supplied to several tools. We analyzed the sentiment of the question title using six different sentiment analysis tools: SentiStrength [12], SentiStrength-SE [6], SentiCR [7], Senti4SD [2], StanfordNLP Sentiment Analyzer [13], and a sentiment analyzer [14] built on top of the Natural Language Toolkit (NLTK) [15].

A. Running Sentiment Tools

SentiStrength is a sentiment analysis tool that specifies the sentiment of short texts and is capable of rating the strength of the text sentiment. The tool uses a lexicon that includes words and word stems with different sentiments and sentiment strengths. SentiStrength-SE builds upon this tool by developing a domain specific dictionary for the tool to make it suitable for analyzing software engineering related text [6].

SentiCR uses NLTK and some machine learning packages to build a sentiment analysis tool that is trained on software engineering domain text. This tool is created to specifically analyze the sentiment of code review comments [7] and is useful for analyzing other software engineering artifacts.

The StanfordNLP Sentiment Analysis tool [13] uses a different approach to improve the understanding of the meaning of longer phrases, using sentiment labels for a vast number of phrases and employing neural networks to build a sentiment analysis tool that is trained on the mentioned labels. The tool is trained on movie reviews and can output a tree structure providing detailed sentiment assignments to each word for each sentence. However, the tool does not give an overall sentiment analysis for a block of text that consists of multiple sentences. We also used a sentiment analyzer implemented using NLTK that was also trained on movie reviews [14]. We also used Senti4SD [2], a sentiment classifier trained on Stack Overflow questions and specifically designed to analyze the sentiment in developers' communication mediums.

We analyzed both the question title text and the combination of question title text and question body. We wanted to compare the results of our analysis with the users' rating of the question sentiment, but the users were not asked about the strength of the sentiment. They were given the options of neutral (0), more positive (+1), and more negative (-1) to choose from when asked about the sentiment of the question. Thus, we had to accordingly adjust the results of two of the sentiment analysis tools to match this rating system. We mapped out our method of converting the sentiment analysis tools' results to a rating system that aligned with our developers' rating. The tools' ratings are converted into either -1, 0, or 1, to indicate negative, neutral, or positive sentiment. Replication package provides more details on the conversion [11].

B. Aggregating Sentiment Scores for each SO question

A single user rating of the question's sentiment was calculated by selecting the category that most participants selected.

TABLE I: The Stack Overflow questions presented to the participants

#	Question ID	Description	Title	Question Word Count	Question Body Word Count	Votes	Tags	Code Snippet Count	Answer Count	Views
1	363681	Generating random integers within a range	How do I generate random integers within a specific range in Java?	12	22	3399	java, random, integer	2	67	3.9M
2	477816	Correct JSON content type	What is the correct JSON content type?	7	100	10034	json, http-headers, content-type	0	36	2.7M
3	1642028	Function of an operator in C++	What is the "–>" operator in C++?	7	60	8658	c++, c, operators, code-formatting, standards-compliance	1	22	758K
4	2003505	Deleting a Git branch	How do I delete a Git branch locally and remotely?	10	33	16467	git, version-control, git-branch, git-push, git-remote	1	39	7.9M
5	5955130	Redirecting a page to other using a Java method	JSF page redirecting from java bean	6	31	36	jsf, redirect, managed-bean	1	3	85K
6	11133760	Converting MySQL date to Unix timestamp	MySQL convert date string to Unix timestamp	7	33	131	mysql, datetime, timestamp, unix-timestamp	1	4	312K
7	1959040	Sending arguments to a JavaScript function	Is it possible to send a variable number of arguments to a JavaScript function?	14	55	161	javascript, variadic-functions	2	12	105K
8	2867675	Table duplication in MySQL	Duplicating table in MYSQL without copying one row at a time	11	65	5	mysql, database, duplicates	0	6	2K
9	2626325	Operand size conflict	Operand size conflict in x86 Assembly?	6	82	7	assembly, x86	0	3	8K
10	776701	Nullable parameters in LinqToSql	LinqToSql stored procedures always makes the parameters nullable. Why?	9	75	1	linq-to-sql, stored-procedures	0	1	277

For example, a question with 1 participant labeling it as neutral, 3 participants labeling it as positive, and 4 participants labeling it as negative, would be labeled as overall a negative sentiment. Ties were labeled neutral.

IV. Pilot Study Results

A. Feature Rating - Revalidating Calefato et al. [1]

In order to see what features developers prefer in questions, we break down the participants selections from our questionnaire. The first key insight we can find is that a majority of developers prefer having more code snippets in every SO question except for Q2. In addition, this was the only feature that every developer selected at least once. Another insight is that developers prefer short titles and bodies as it was the second most selected feature followed by multiple tags. There was less consensus on selection of the multiple tags attribute. Four developers selected this attribute on most questions while 4 never selected this feature.This is in contrast to the Multiple URLs attribute had only 2 developers who never selected it. Finally, the lack of uppercase characters was only brought up by two developers who selected this attribute on 8 and 2 of the questions respectively. In summary, Calefato et al. [1] reported top features were adding code snippets, using multiple tags, and including URLs, whereas in our study they were code snippets, short titles, and multiple tags; giving us a two feature overlap with [1].

B. RQ1 Results: Developer Rating of SO Question Sentiment

Figure 1 shows how the participants ranked the questions' sentiment. Two participants ranked the sentiments of the answers that were written to the question, so we did not include their rating in the chart. The results indicate that questions 1, 3, 5, 6 were mostly ranked neutral, while questions 2 and 7 were mostly ranked positive and questions 8 and 10 were mostly ranked negative. Question 9 and 4 received polarizing descriptions, since the number of participants who believed that the sentiment is positive was equal to those who found it negative. As there isn't a clear indication whether the sentiment was neutral or positive, tied questions are considered neutral for our analysis. The Light's Kappa of our study's raters was 0.101, indicating developers had very different views about the sentiment of questions overall.

Looking further into a few of the questions, we can see some interesting patterns. First, we can see from the results of Question 8 that small decisions about word choice and grammar can play a large role in its sentiment. Of the 4 participants who rated this question as negative, two of them specifically called out the word 'painful' in the original post with one participant saying "Words like 'painful' give a negative tone". In the context of the post, the user is trying to duplicate a table and avoid copying it row by row and says "row by row insert is very painful (because of 120M rows)." The other 2 participants who rated this question as negative seemed to comment on the overall syntax of the question. One participant said "All of the surrounding phrases seemed negative" in regards to the original user stating that they do not want to copy the table row by row. The other participant said "Stuff was written with bad grammar and it was too short which made it seem harsher." It is possible that not all the tools agreed with the user rating sentiment because the negative sentiment of this post was based on the negative connotation of a single word, "painful". In the results of Question 10, 5 participants who rated this question as having negative sentiment, 3 participants directly mentioned the user's usage of the :(emoticon. Finally, it is intriguing to note that some of the participants interpreted curiosity, clarity, and excitement as attributes that indicate positive sentiment of text.

C. RQ2 Results: Comparing Tools with Developer Sentiment

Table II shows the result of sentiment analysis tools on the question titles, and Table III shows the result of sentiment analysis tools on the question title text and question body with the aggregated developer rating shown in the last column. If a tool had the same sentiment ratings for both the question titles only and the question title text and question body, the entry in Table III has an asterisk to indicate this agreement on the sentiment of the question title text and the combination of

TABLE II: Results on SO Question Title Text. “-1”- negative sentiment, “0”- neutral sentiment, and “1”-positive sentiment

Question	SentiStrength	SentiStrength-SE	SentiCR	NLTK	Stanford NLP	Senti4SD	Developer Rating
1	0	0	0	-1	0	0	0
2	0	0	0	1	0	0	1
3	0	0	0	0	-1	0	0
4	0	0	0	-1	-1	0	0
5	0	0	0	-1	0	0	0
6	0	0	0	0	-1	0	0
7	-1	0	0	-1	0	0	1
8	0	0	0	-1	0	0	-1
9	0	0	0	1	-1	0	0
10	0	0	-1	1	-1	0	-1

TABLE III: Results on SO Question Title Text and Question Body. “-1”-negative sentiment, “0”-neutral sentiment, and “1”-positive sentiment. * indicates sentiment of SO Question Title Text and Question Body == Question Title Text.

Question	SentiStrength	SentiStrength-SE	SentiCR	NLTK	Stanford NLP (Average and By Sentence)	Senti4SD	Developer Rating
1	0*	0*	-1	-1*	0* [0, 0]	0	0
2	-1	-1	-1	-1	-0.33 [0, -1, 0, -1, 0, 0]	1	1
3	0*	0*	0*	1	-0.5 [-1, -1, 0, -1, -1, 0, +1, -1]	0	0
4	0*	0*	0*	-1*	-0.75 [-1, -1, -1, 0]	1	0
5	0*	0*	0*	-1*	0* [0, 0, 0, 0, 0]	0	0
6	0*	0*	0*	-1	-0.75 [-1, 0, -1, -1]	0	0
7	1	1	0*	1	0.25 [0, 0, +1, 0]	0	1
8	-1	0*	-1	-1*	-0.83 [-1, -1, -1, -1, -1, 0]	-1	-1
9	-1	-1	0*	-1	-1* [-1, -1, -1, -1, -1, -1]	-1	0
10	-1	-1	-1*	-1	-0.625 [1, -1, -1, 0, 0, -1, 0, -1]	-1	-1
Agreement with developer rating	8/10	7/10	7/10	3/10	4/10	7/10	

question title text and question body. To analyze the combination of title text and question body, we calculated the average of the ratings for each sentence. The average ratings are shown in Table III. Overall, the sentiment analysis on the question titles and text, as shown in Table III, is more aligned with the average user rating. Even though SentiStrength’s results are the closest to the developer ratings (the same rating on 8 out of 10 questions), on average the tools that are trained using Software Engineering related text (SentiStrength-SE, SentiCR, and Senti4SD) performed better than general purpose tools and gave closer ratings to user ratings (7 out of 10 questions for each tool). Developers cited excitement, willingness to learn, humility, directness, and casual language as reasons why they interpreted the question as positive in sentiment.

V. Conclusions and Future Work

The paper presents results of a pilot survey on how developers perceive sentiment in SO questions and how tool output compares to their perceived sentiment. Six questions were rated by developers as neutral, two questions were rated positive, and two were rated negative. On average, tools designed to detect sentiment of SE texts performed better and were more closely aligned with developer ratings compared to general purpose tools. Future work plans include an in-depth qualitative analysis to better explain the discrepancies between tools and developer ratings.

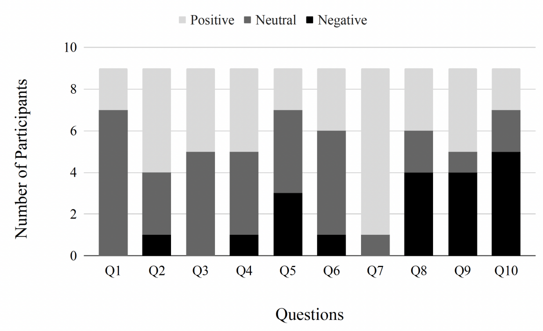

Fig. 1: Participant ranking of the sentiment of each question

References

[1] F. Calefato, F. Lanubile, and N. Novielli, “How to ask for technical help? evidence-based guidelines for writing questions on stack overflow,” *IST Journal*, vol. 94, pp. 186–207, 2018. [Online]. Available: https://doi.org/10.1016/j.infsof.2017.10.009

[2] 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. [Online]. Available: https://doi.org/10.1007/s10664-017-9546-9

[3] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto, “Sentiment analysis for software engineering: How far can we go?” in *Proc. of ICSE*, ser. ICSE ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 94–104. [Online]. Available: https://doi.org/10.1145/3180155.3180195

[4] N. Novielli, D. Girardi, and F. Lanubile, “A benchmark study on sentiment analysis for software engineering research,” in *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*. IEEE, 2018, pp. 364–375.

[5] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, “Sentiment polarity detection for software development,” *Empirical Softw. Engg.*, vol. 23, no. 3, p. 1352–1382, Jun. 2018. [Online]. Available: https://doi.org/10.1007/s10664-017-9546-9

[6] M. R. Islam and M. F. Zibran, “Sentistrength-se: Exploiting domain specificity for improved sentiment analysis in software engineering text,” *JSS*, vol. 145, pp. 125–146, 2018.

[7] T. Ahmed, A. Bosu, A. Iqbal, and S. Rahimi, “Senticr: a customized sentiment analysis tool for code review interactions,” in *(ASE)*. IEEE, 2017, pp. 106–111.

[8] N. Imtiaz, J. Middleton, P. Girouard, and E. Murphy-Hill, “Sentiment and politeness analysis tools on developer discussions are unreliable, but so are people,” in *2018 IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. IEEE, 2018, pp. 55–61.

[9] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto, “Sentiment analysis for software engineering: How far can we go?” in *Proceedings of the 40th International Conference on Software Engineering*, 2018, pp. 94–104.

[10] R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik, “On negative results when using sentiment analysis tools for software engineering research,” *Empirical Software Engineering*, vol. 22, no. 5, pp. 2543–2584, 2017. [Online]. Available: https://doi.org/10.1007/s10664-016-9493-x

[11] N. Mansoor, C. Peterson, and B. Sharif, “Replication Package for How Developers and Tools Categorize Sentiment inStack Overflow Questions - A Pilot Study,” Mar. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4602645

[12] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, “Sentiment strength detection in short informal text,” *Journal of the American society for information science and technology*, vol. 61, no. 12, pp. 2544–2558, 2010.

[13] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[14] “Sentiment analysis,” http://text-processing.com/docs/sentiment.html, accessed last: 01/29/2020.

[15] E. Loper and S. Bird, “Nltk: the natural language toolkit,” *arXiv preprint cs/0205028*, 2002.