Sentiment of Technical Debt Security Questions on Stack Overflow: A Replication Study

Jarl Jansen
Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
j.l.o.jansen@student.tue.nl

Nathan Cassee
Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
n.w.cassee@tue.nl

Alexander Serebrenik
Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
a.serebrenik@tue.nl

***Abstract*—Technical debt (TD) refers to the accumulation of negative consequences resulting from sub-optimal solutions during software development. A recent paper by Edbert *et al.* studied the difference between security-related TD questions, and security-related non-TD questions on Stack Overflow (SO).**

One of the characteristics under investigation is the sentiment expressed in these two categories as sentiment provides insight into developers' attitudes and emotions toward security-related TD. To this end, Edbert *et al.* used a general-purpose, off-the-shelf, sentiment analysis tool. However, previous research has shown that general-purpose off-the-shelf sentiment tools are potentially unreliable when applied to software engineering texts. Therefore, we replicate the study by Edbert *et al.* using state-of-the-art sentiment analysis tools purpose-built and fine-tuned on SE data, to understand whether and how tool-choice influences the obtained results. We consider both shallow (Senti4SD) and deep learning (BERT4SentiSE) tools.

To further understand the differences between shallow and deep-learning sentiment analysis tools, we perform a qualitative analysis into the underlying reasons for tools disagreement. We identify five categories of disagreements: misunderstanding context, courtesy phrases, subjective sentiment, brevity, and divergent examples.

Our results are relevant to academics, reiterating the relevance of careful selection of tools used to perform sentiment analysis. Furthermore, the results are relevant to users and developers of sentiment analysis tools, as they inform tool selection dependent on the application domain, and provide insight into optimization of the pre-processing steps.

Finally, our study shows that retraining sentiment analysis tools with identical data fails to resolve fundamental inconsistencies between how certain types of language, such as courtesy phrases, are classified.

***Index Terms*—Replication study, Sentiment Analysis, Security, Technical debt, Stack Overflow**

I. Introduction

Technical debt (TD) is a metaphor representing the accumulated negative consequences resulting from choosing expedient or sub-optimal solutions during software development [11]. TD can result in negative consequences such as increased complexity, increased vulnerabilities, and reduced maintainability [20].

To manage software security in the development life cycle the concept of TD has been extended to the security domain, thereby introducing the notion of security-related TD [28]. Security-related TD is TD that results in sub-optimal security practices. These practices can weaken the security of a system significantly and potentially result in exploitable vulnerabilities [19], hence managing security TD is of the essence.

To obtain further insights into the challenges and needs surrounding security-related TD Edbert *et al.* [14] have recently studied security-related TD questions (STDQs) at Stack Overflow (SO). SO is an extensive archive of SE knowledge, offering information on specific technologies and corresponding developer perspectives [3], and has been used to study TD in the past [2, 16].

One of the aspects studied by Edbert *et al.* [14] was the sentiment of STDQs on SO. Analyzing sentiment improves understanding of popularity and emotion toward security-related TD questions [22]. The study by Edbert *et al.* found that the sentiment expressed by security-related TD SO questions is mostly neutral. Furthermore, the sentiment expressed in STDQs is comparable to the sentiment expressed by security-related non-TD SO questions.

To perform this analysis, Edbert *et al.* used the VADER sentiment analysis tool from the NLTK package. However, Lin *et al.* [21] have shown that off-the-shelf sentiment analysis tools such as VADER perform poorly on SE data, and hence Lin *et al.* discourage the usage of tools such as VADER within a SE context. Furthermore, they recommend using sentiment analysis tools that have been retrained on SE data when conducting sentiment analysis within the SE domain.

Given the relevance of security-related TD and the potential inaccuracies of off-the-shelf sentiment analysis tools, we have opted to conduct an independent replication [9, 12, 32] of the sentiment analysis study of Edbert *et al.* [14]. Specifically, when replicating the study of Edbert *et al.* [14] we are interested in finding whether replacing VADER with SE-specific sentiment analysis tools such as Senti4SD [7] and BERT4SentiSE [6] would affect the study conclusions. We opt not to include state-of-the-art Large Language Models, as they are currently infeasible to run on large data sets. Through replication we answer the following research questions:

- *RQ1: What sentiment is expressed in security-related technical debt questions on Stack Overflow?*
- *RQ2: How does the sentiment contrast with the sentiment of non-technical debt security-related questions on Stack Overflow?*

Building on our replication study we further reflect on similarities and differences between shallow-learning and deep-learning SE-specific sentiment analysis tools. While previous research has investigated the difference in performance between such tools [33], the question arises in what context a specific tool is most appropriate, and why. Hence, we conduct a follow-up qualitative study to better understand why shallow and deep-learning tools disagree, i.e., answer

- *RQ3: What are the underlying reasons as to why Senti4SD and BERT4SentiSE evaluate a SO question to have a different sentiment?*

The remainder of this paper is organized as follows. In Section II we discuss related literature, and in particular SE-specific sentiment analysis tools Senti4SD and BERT4SentiSE. In Section III we discuss the methodology employed for our analysis. This is followed by a presentation of our findings in Section IV. Threats to validity are presented in Section V. Section VI contains a discussion of presented results, Section VII the implications of the research, and Section VIII concludes.

II. Related Work

Sentiment analysis tools have been extensively used to analyze software engineering data. For example, Calefato *et al.* [8] observed that successful SO questions typically employ a neutral emotional tone. When considering the security domain specifically, Pletea *et al.* [26] have previously conducted research into the sentiment of security-related discussions on GitHub. They found that these discussions compromise approximately 10% of the total discussions on GitHub. The sentiment of security-related discussions was more negative than non-security-related discussions. While this work used NLTK VADER, an off-the-self sentiment analysis tool, and hence *a priori* its results might be inaccurate, they have been confirmed by subsequent replication studies [24].

a) SE-specific sentiment analysis tools: Shallow-learning tool Senti4SD was originally introduced by Calefato *et al.* [7]. Senti4SD is a distributional semantic model (DSM) and uses both lexicon and keyword-based features, as well as word embeddings to obtain semantic features. Senti4SD was originally trained on the gold standard SO data-set introduced in that same paper. SentiCR is another shallow-learning tool developed by Ahmed *et al.* [1]. SentiCR converts the input text into a vector using a bag-of-words approach. Then for classification, the Gradient Boosting Tree (GBT) algorithm is applied to the vector.

Deep-learning tool BERT4SentiSE is a supervised deep-learning tool originally introduced by Biswas *et al.* [6]. BERT4SentiSE is based on the BERT model developed at Google [13]. The tool uses a language representation model to effectively answer natural language processing tasks. Introduced by Chen *et al.* [10], SentiMoji is a deep-learning tool built atop DeepMoji [15]. DeepMoji has learned using Twitter and Github data to classify sentiment by associating emojis with text. Emojis representing the text are then transformed into a vector, which in the final layer of SentiMoji is used to classify sentiment polarity.

b) Tool benchmarking: Novielli *et al.* [23] have looked at the performance of SE-specific sentiment analysis tools and their accuracy in different settings. In particular, they looked at lexical-based and supervised sentiment analysis tools. Supervised models are trained on SE data such as SO questions, GitHub discussions, or Jira issues. The performance of supervised models is significantly better in a within-platform setting, meaning the tools are better at evaluating samples originating from the same platform as their training set. Further work by Uddin *et al.* [33] confirmed these results and added deep learning SE-specific sentiment analysis tools to the comparison. In the within-platform setting for SO data the deep learning tool BERT4SentiSE performs best on all three of the evaluation metrics (precision 0.88, recall 0.88, F1 score 0.88). For shallow ML-based tools, Senti4SD performs best (precision 0.85, recall 0.85, F1 score 0.85).

Zooming in on the misclassifications of the shallow-learning tools Novielli *et al.* [25] identified seven categories of misclassifications. The most common category was *polar facts*, phrases that evoke an emotion while the text remains neutral. *General errors* are cases where the tool is unable to cope with the context or misclassifies because of poor pre-processing. *Politeness*, are instances where tools struggle to differentiate between neutral and non-neutral sentiment. In *implicit sentiment polarity*, emotion is not expressed explicitly through emotive words. *Subjectivity in sentiment analysis*, are cases where the evaluation of sentiment is subjective. Lastly, the *inability to deal with pragmatics or context information*, and *figurative language* were the two least common categories.

c) Replications of SE-specific sentiment analysis studies: Jongeling *et al.* [18] and Novielli *et al.* [24] have conducted replication studies of sentiment in software engineering texts. Jongeling *et al.* replicated two empirical SE studies that use off-the-shelf sentiment analysis tools. In their replication study, they used four off-the-shelf sentiment analysis tools. When replicating the study by Pletea *et al.* [26] vastly different data was obtained, yet Jongeling *et al.* were able to confirm most of the conclusions of Pletea *et al.*. For the second study Jongeling *et al.* replicated, the conclusions could not be confirmed. Novielli *et al.* [24] also replicated the work of Pletea *et al.* using 4 SE-specific sentiment analysis tools. The original conclusions were again mostly valid despite the tools obtaining dissimilar distributions of sentiment. For the second work replicated by Novielli *et al.* the conclusions could not be validated, for three SE-specific tools partially contradictory conclusions were obtained.

III. Methodology

As befitting a replication study we follow the methodology employed in the original work by Edbert *et al.* [14]. The only differentiation from the original study is the selection of sentiment analysis tools. We re-use the dataset from the

original work by Edbert *et al.* [14]. To classify the sentiment of each question we follow the procedure used by Edbert *et al.* [14] to preprocess the data, and we append the title to the body for each SO question. Then we apply three sentiment analysis tools to the resulting data set: VADER, the tool used by Edbert *et al.*, Senti4SD [7], and BERT4SentiSE [6]. To answer RQ1 and RQ2, we perform statistical analysis of the distributions of sentiment values reported by the tools, while for RQ3, we perform thematic analysis of the disagreement between the tools.

A. Data

Using the predefined list of SO tags identified by Yang *et al.* [34], Edbert *et al.* [14] have collected SO questions tagged with such security-related tags as "sql-injection" or "websecurity". Next, Edbert *et al.* used an ML classifier to categorize the questions into technical debt, i.e., STDQ, and non-technical debt, i.e., non-STDQ. The dataset contains 45,078 (38%) STDQs and 72,155 (62%) non-STDQs. We do not replicate this classification process and consider the same 45,078 STDQs and 72,155 non-STDQs as in the original study by Edbert *et al.*

B. Sentiment Analysis Tools

VADER represents the baseline against which we can compare the other SE-specific sentiment analysis tools. VADER outputs a decimal score between -1 (negative) and 1 (positive). We convert this decimal score to a ternary score using the procedure outlined in VADER's documentation[1]: A score greater than or equal to 0.05 is mapped to "positive," score smaller than or equal to -0.05—to "negative," and a score between -0.05 and 0.05— to "neutral."

We compare VADER with the best-performing shallow-learning and the best-performing deep-learning tools on SO data [33], namely Senti4SD [7] and BERT4SentiSE [6]. Since Senti4SD is pre-trained using the gold standard SO sentiment data set introduced by Calefato *et al.* [7] we use the default hyperparameters for Senti4SD within our study. We have retrained BERT4SentiSE using the same gold standard SO data set. Both Senti4SD and BERT4SentiSE classify input text as either "positive", "negative" or "neutral."

C. Analysis

To answer RQ1, we plot the proportion of SO questions corresponding to each tool's three polarity categories (negative, neutral, positive). To answer RQ2, we test whether the sentiment distribution between STDQs and non-STDQs differs using the Cochran-Armitage test. Under the assumption that all three sentiment polarity classes can be ordered (*positive* $>$ *neutral* $>$ *negative*). We use the traditional significance threshold of 0.05 and to control for multiple comparisons we correct the p-values using the Benjamini-Hochberg procedure [4].

To answer RQ3 we perform a qualitative analysis of disagreement between the tools. Inspired by grounded theory building [30] we follow an iterative approach. We sampled random batches of 20 SO questions that were classified differently by Senti4SD and BERT4SentiSE. For each question in the batch, the sentiment is manually evaluated using the guidelines provided by Calefato *et al.* [7] based on the framework of Shaver *et al.* [31]. To ensure consistency between batches, the same author evaluates each batch. Having manually established the sentiment of the question, we next determine the phrases that influenced the classifiers in their sentiment evaluation and group the phrases into broader categories. Batches are sampled until no new categories are obtained, i.e., saturation is reached.

D. Availability of Data

To encourage further replications of our study all study material produced, such as input data, generated data, and code has been made available in the replication package.[2]

IV. Results

A. ***RQ1:*** *What sentiments are expressed in security-related technical debt questions on Stack Overflow?*

Table I and the right column of Figure 1 summarize the numbers of positive, neutral and negative security-related STDQs as identified by VADER, Senti4SD, and BERT4SentiSE. We note that VADER rates almost all SO questions as non-neutral, and the majority as positive. Both Senti4SD and BERT4SentiSE predominantly rate the questions with a neutral sentiment. This observation is aligned with the observation of Raman *et al.* [27] that many terms such as 'abort' and 'kill' that have negative connotations in general English, but are neutral in software engineering. The numbers of questions Senti4SD and BERT4SentiSE label as negative and positive are relatively similar.

The paper by Edbert *et al.* states that STDQs "typically have a neutral sentiment" [14]. Table I invalidates this conclusion when VADER is considered, the very same tool used by Edbert *et al.* in the original study. The conclusion is, however, valid when considering SE-specific tools Senti4SD and BERT4SentiSE. Since SE-specific tools have been repeatedly shown to be a better proxy for human sentiment assessment, we believe that the statement that STDQs "typically have a neutral sentiment" has been confirmed.

RQ_1

STDQs have a positive sentiment when VADER is used for analysis contradicting the results of Edbert et al.*; SE-specific tools Senti4SD and BERT4SentiSE find that STDQs are mostly neutral.*

[1]https://github.com/cjhutto/vaderSentiment

[2]https://tinyurl.com/4n86bnbb

TABLE I
SENTIMENT OF SECURITY-RELATED TD QUESTIONS ON SO

Tools	Expressed sentiment		
	Positive	Neutral	Negative
VADER	33,904	1,517	9,657
Senti4SD	11,937	19,626	13,515
BERT4SentiSE	4,708	35,879	4,491

B. ***RQ2:*** *How does the sentiment contrast with the sentiment of non-technical debt security-related questions on Stack Overflow?*

There are 72,149 non-STDQs, i.e., non-TD security-related SO questions in the data set. The number of questions corresponding to each of the three polarity labels is displayed in Table II and the left column of Figure 1. Similarly to STDQs, we find that VADER rates the majority of questions positive, and very few neutral. The distribution of sentiment of Senti4SD and Bert4SentiSE for non-TD questions seems to be very similar to the distribution of TD questions. Again most questions are rated neutral.

TABLE II
SENTIMENT OF SECURITY-RELATED NON-TD QUESTIONS ON SO

Tools	Expressed sentiment		
	Positive	Neutral	Negative
VADER	45,725	7,062	19,362
Senti4SD	18,379	31,911	21,859
BERT4SentiSE	7,808	58,785	5,556

To verify whether the distribution of sentiment is statistically different between TD questions and non-TD questions we run the Cochran-Armitage statistical test. The resulting statistic and p-values (rounded to 4 decimals) can be found in Table III.

As outlined previously we use a significance level of $p < 0.05$ in our statistical tests and we corrected the p-values using the Benjamini-Hochberg [4] procedure to control for multiple comparisons. For the Cochran-Armitage test, we obtain p-values of less than 0.05 for all three tools. Hence we can reject the null hypotheses, meaning the distributions of sentiment for STDQs and non-STDQs are different.

Edbert *et al.* concluded that the sentiment in security-related TD questions is comparable to the sentiment expressed in security-related non-TD questions [14]. Overall, the shapes of the distributions in the left column of Figure 1 are similar to the shapes of the corresponding distributions in the right column, providing some support to the conclusion by Edbert *et al.* However, statistical analysis reveals that the distributions of sentiment STDQs and non-STDQs are statistically different.

TABLE III
STATISTICAL TEST ON DISTRIBUTION OF SENTIMENT BETWEEN STDQS VS NON-STDQS. ZERO AS THE P-VALUE MEANS THAT THE P-VALUE IS TOO SMALL TO BE COMPUTED EXACTLY.

Tools	Cochran-Armitage	
	statistic	p-value
VADER	2465	0
Senti4SD	14.81	0.0006
BERT4SentiSE	181.54	0

TABLE IV
MANUAL EVALUATION MISS-CLASSIFICATION FREQUENCY

Category	Frequency
Misunderstanding Context	43
Courtesy Phrases	37
Subjective Sentiment	13
Brevity of text	7
Divergent examples	3
No category	**32**

RQ_2

We conclude that the distribution of sentiment for STDQs and non-STDQs is different, contradicting the conclusion by Edbert et al.

C. ***RQ3:*** *What are the underlying reasons as to why Senti4SD and BERT4SentiSE evaluate a SO question to have a different sentiment.*

We manually evaluated 6 batches of 120 security-related SO questions at which point saturation was reached. We detected the following five categories for causes of misclassification: courtesy phrases, misunderstanding context, brevity, divergent examples, and subjective sentiment.

Courtesy phrases are words and phrases that express politeness but are not necessarily emotionally charged, e.g., “thanks” or “help is appreciated”. *Misunderstanding context* refers to language that should be considered neutral within the specific communicative context but which might be misinterpreted as negative or positive if the context is ignored. *Brevity* refers to short text fragments ($<$ 20 words) that are neutral but result in at least one of the tools evaluating the fragment as positive or negative. *Divergent examples* refer to examples given by the author of the SO question that are inherently confusing, these include code as well as emotive dummy text. One such example is ”Here is a hint: ’GOOD’ ’LUCK’, it ’SOUNDS SIMPLE TO ME’”. Lastly, *subjective sentiment* are fragments where the sentiment is not clear. These can be cases where both overtly positive and negative sentiment is expressed, or in general, when a reasonable argument for more than 1 sentiment label can be made. Any SO question can be assigned zero or more of these categories.

During the manual evaluation of 120 SO questions, we found 43 cases where misunderstanding context occurred, 37 cases using courtesy phrases, 13 cases with subjective sentiment, 7 cases where brevity of text was relevant, and 3

cases where divergent examples played a role. See Table IV for the results. For 32 questions (27%) we could not determine any category accurately explaining the miss-classification. An example of such an inexplicable question would be the following:

> *Find out map path from user account to Security group[Win server 2012].*
> *I have system account which is part of a security group. the account is added to SG indirectly. How can i find the map path between the user account and Security group*

This fragment clearly expresses a neutral sentiment, nonetheless, Senti4SD assigned a positive polarity to this fragment. None of the hypotheses that could potentially explain the positive evaluation in this case were consistent with the other fragments we analyzed, and hence we leave these inexplicable questions uncategorized.

Courtesy phrases seem to have a noticeable effect on BERT4SentiSE. In total, there are 37 questions in which courtesy phrases occur. BERT4SentiSE rates 16 of these questions with a positive sentiment. In total, BERT4SentiSE only rates a total of 17 questions as positive, hence the vast majority of positively rated questions by BERT4SentiSE contain courtesy phrases. Questions containing courtesy phrases are often rated as positive by BERT4SentiSE despite being neutral (10 such cases in our manually annotated sample), hence courtesy phrases result in many false positives for BERT4SentiSE. BERT4SentiSE only rates a question with courtesy phrases as negative if there is some other overtly negative expression in the question which is why these classifications tend to be accurate. Meanwhile, Senti4SD does not seem to be affected by courtesy phrases, out of the 37 questions containing courtesy phrases Senti4SD only classifies 12 as positive, and 17 as negative. Therefore it seems that Senti4SD ignores courtesy phrases in its polarity assessment while BERT4SentiSE tends to use it as an indication of positive polarity. Questions containing courtesy phrases often result in opposite classifications by Senti4SD and BERT4SentiSE, here, opposite classifications occur when one classifier rates the fragment as positive, and the other as negative. Out of the 13 questions with opposite classifications, 11 contain courtesy phrases.

Deep-learning tools are expected to outperform shallow-learning tools when understanding of context is important in classifying a fragment. In total there are 43 SO questions in our analysis for which misunderstanding of context leads to misclassifications. BERT4SentiSE correctly classifies 35 out of 43 such cases. While Senti4SD correctly classifies only 5 out of 43 cases. This seems to confirm our hypothesis that deep-learning tools are better at understanding context.

Brief SO questions ($<$ 20 words) do not seem to affect BERT4SentiSE, for all 7 instances in our data-set BERT4SentiSE correctly labels them as neutral. Senti4SD rates all 7 of these instances as either positive or negative, hence Senti4SD incorrectly classifies all SO questions of short length. A potential reason for this is that Senti4SD detects some word or phrase with a slight polarity, since the text fragment is short this polarity dominates the sentiment calculation, causing the fragment to be rated incorrectly.

The sample contains 3 SO questions in which the examples used confuse the tools. Code examples that were not contained in an HTML or markdown element have not been removed during pre-processing. These cases seem to confuse both classifiers. Examples that contain words with emotional polarity confuse BERT4SentiSE, however, the sample size is insufficiently small to verify the effects precisely.

There are 13 instances where confusion arose during the manual labeling of sentiment. For 3 of these cases, the tools assigned opposite sentiment classifications.

RQ_3

We found five different causes for missclassifications between BERT4SentiSE and Senti4SD. Notably, BERT4SentiSE is more likely to classify text containing courtesy phrases as positive and is more accurate when the expression of sentiment is context-dependent. Meanwhile, Senti4SD incorrectly classifies short text as non-neutral.

V. Threats to validity

Our replication study adheres closely to the methodology of the original work; consequently, several threats to validity that are pertinent to the original study are also applicable to this replication.

Construct validity refers to the degree to which a measurement or test accurately assesses the concept it intends to measure. Similar to the original work by Edbert *et al.* the data set of SO questions was filtered using the keywords associated with an SO question. This could lead to inconsistencies as the keywords are assigned by the author of the SO question. The original paper's authors manually checked a statistically significant sample and determined that in 97% of cases, the questions were indeed security-related. Our study uses the same data set, and thus, this is also applicable to our study. Manual verification reduces the threat to validity; however, a potential threat persists due to the subjectivity of manual evaluation. Furthermore, during filtering, only the 9 keywords as identified by Yang *et al.* [34] were used. Security questions not using these keywords could have been missed, resulting in a potential threat to validity.

The qualitative analysis in our study used a manual evaluation of SO questions. Emotion perception, which includes sentiment evaluation, is subjective to the person conducting the evaluation [29]. The manual evaluation used in the qualitative analysis was conducted by a single author, thereby potentially introducing subjectivity into the assessment. To mitigate this threat, we excluded difficult cases from our analysis by not assigning any sentiment to them in our manual labeling. Secondly, in line with recommendations [23] we used the emotion classification framework of Shaver *et al.* [31].

Internal validity is the extent to which a study accurately measures the impact of the independent variable. The classifier determining whether SO questions are TD or non-TD forms a threat to the study's internal validity. The classifier had an F1 score of 0.75 in the original work, which is sub-optimal as the data set may contain false positives and false negatives.

The completeness and accuracy of the misclassification categories identified in our qualitative analysis can not be verified, as they are exploratory. Some of the categories do conform with a previous qualitative analysis that compared misclassifications of several shallow-learning tools [25], indicating our results are not completely unfounded.

External validity is the extent to which a study can be generalized outside the study setting. Similar to the original study, we restrict our analysis to SO data; generalizing our conclusions to security-related TD in general is not necessarily valid. By making the materials used in this study publicly available, we encourage evaluation of its external validity.

Furthermore, we aim to generalize our qualitative analysis to shallow and deep-learning tools in general. This generalization is not necessarily valid, as the shallow-learning and deep-learning tools not evaluated in our study use different training data and use comparable but different classification techniques.

Conclusion validity is the extent to which the inferences and conclusions that are drawn are warranted. In the original study by Edbert *et al.*, conclusions are rather vague, using terms such as comparable to describe how the distribution of sentiment among STDQs and non-STDQs differ. In our replication study, we precisely define and verify hypotheses using appropriate statistical tests, thereby reducing the threat to conclusion validity. Additionally, we minimize the false discovery rate by controlling for multiple comparisons.

VI. Discussion

It is important to note that when replicating the original study, with the same tool, using the same data, we do not conform the conclusion of Edbert *et al.* [14] that STDQs are mostly neutral. Meanwhile, when we use SE-specific sentiment analysis tools, we do find that STDQs are mostly neutral. This results in a curious situation, where the original study's data does not support the drawn conclusion, yet the data derived using an independent replication does confirm their conclusion.

The original paper also claims that the sentiment in STDQs and non-STDQs are comparable. Using statistical tests, we find that this conclusion can not be validated for sentiment derived using VADER, Senti4SD, or BERT4SentiSE. This shows the importance of using hypothesis tests, as opposed to merely relying on visual confirmation. Furthermore, this suggests that developers experience dealing with security-related TD differently from dealing with security-related non-TD. This discrepancy could be caused by factors such as lack of understanding, usefulness in obtaining short-term benefit, and frustration [5, 20].

Previous replication studies in sentiment analysis for software engineering have sought to verify claims about a singular polarity label when investigating the effect of SE-specific sentiment analysis tools on conclusion validity [18, 24]. In other words, the replications studied the validity of conclusions claiming that a certain data group is more negative/positive/neutral than a different group. In this work, we verified a similar claim: STDQs are mostly neutral. However, we also observed that the polarity distribution differs between BERT4SentiSE and Senti4SD. Hence, conclusions that make a claim about a singular polarity label are weaker than conclusions about the general distribution of polarity between groups. In contrast to previous replication studies, we also verified a stronger claim: Specifically about the general distribution of sentiment between STDQs and non-STDQs. We found that sentiment polarity distributes differently across these two categories. The original work by Edbert *et al.* [14] claimed the distributions are comparable, further highlighting that claims about a specific polarity label tend to be easier to validate than claims about the general distribution.

Further inspection of disagreements between Senti4SD and BERT4SentiSE indicates that misunderstanding context, courtesy phrases, and subjective sentiment each influence misclassification. These findings seem to correspond with previous research analyzing the misclassifications of shallow-learning tools [25]. Misunderstanding context seems to occur frequently in our study. Deep-learning tools such as BERT4SentiSE use a language representation model to process natural language effectively [6]. This method seems to result in a better understanding of contextual semantics than shallow-learning tools, such as Senti4SD, which derive their contextual understanding from training data. Furthermore, despite both Senti4SD and BERT4SentiSE having been retrained using the same gold standard SO data set, they evaluate courtesy phrases very differently. Hence, we conclude that merely using the same data set is insufficient for tool consistency. Therefore, when selecting tools these discrepancies should be taken into account to ensure that the chosen tool reflects the desired interpretation of sentiment. Furthermore, this seems to imply that deliberate strategies are necessary to deal with of inconsistencies among tools.

VII. Implications

Below, we summarize the implications of our research for researchers, practitioners, and developers of sentiment analysis tools.

For researchers. We have seen that deep-learning tools such as BERT4SentiSE are better at determining sentiment in situations where context is highly relevant. Future research could investigate whether incorporating more contextually diverse training data for shallow learning tools such as Senti4SD significantly improves the ability to differentiate neutral contexts. Furthermore, the suitability of BERT4SentiSE for situations where context is relevant should be used to inform tool choice. Additionally, the different handling of courtesy phrases by

both BERT4SentiSE and Senti4SD could also be a factor in picking one tool over the other.

We believe that future work should investigate the cause of the discrepancy in sentiment polarity between STDQs and non-STDQs; this information could help managers and/or educators in taking recourse to avoid unnecessary negative sentiment that adversely affects developers in operating [17].

For developers of sentiment analysis tools. As sentiment analysis tools are inconsistent in their evaluation of courtesy phrases developers of sentiment analysis tools should take extra care to ensure the polarity of courtesy phrases conforms to their understanding of courtesy phrases polarity. Especially because we have seen that retraining tools on the same data set does not immediately result in consistency between tools. Therefore, strategies must be developed to address these inconsistencies explicitly, as retraining is unsatisfactory. Developers could also aid accurate tool usage by explicitly stating how their tool deals with courtesy phrases and whether they are considered neutral or non-neutral.

VIII. Conclusion

In this replication study, we investigated the sentiment of STDQs, and how their sentiment contrasts with that of non-STDQs on SO using SE-specific sentiment analysis tools. We validated the claim of the original study by Edbert *et al.* [14] that STDQs are mostly neutral. Furthermore, we investigated the sentiment expressed in STDQs and non-STDQs. We found that their distribution is different, contradicting the assertion in the original work that the sentiment of TD and non-TD security-related SO questions are comparable.

Novel insights were obtained when we further investigated why state-of-the-art shallow and deep-learning SE-specific sentiment analysis tools classify SO questions differently. We found that the deep-learning tool BERT4SentiSE is better at understanding neutral contextual semantics. Furthermore, we found that shallow-learning and deep-learning tools that have been trained on the same data evaluate courtesy phrases fundamentally different, resulting in inconsistent classifications between tools. We therefore recommend careful analysis of the application-domain before tool selection.

References

[1] Toufique Ahmed, Amiangshu Bosu, Anindya Iqbal, and Shahram Rahimi. SentiCR: A customized sentiment analysis tool for code review interactions. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 106–111, 2017.

[2] Reem Alfayez, Yunyan Ding, Robert Winn, Ghaida Alfayez, Christopher Harman, and Barry Boehm. What is asked about technical debt (TD) on Stack Exchange question-and-answer (Q&A) websites? An observational study. *Empirical Software Engineering*, 28(2):35, Jan 2023.

[3] Anton Barua, Stephen W. Thomas, and Ahmed E. Hassan. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, 19(3):619–654, Jun 2014.

[4] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[5] Terese Besker, Hadi Ghanbari, Antonio Martini, and Jan Bosch. The influence of technical debt on software developer morale. *Journal of Systems and Software*, 167:110586, 2020.

[6] Eeshita Biswas, Mehmet Efruz Karabulut, Lori Pollock, and K. Vijay-Shanker. Achieving Reliable Sentiment Analysis in the Software Engineering Domain using BERT. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 162–173, 2020.

[7] Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. Sentiment Polarity Detection for Software Development. *Empirical Software Engineering*, 23(3):1352–1382, Jun 2018.

[8] Fabio Calefato, Filippo Lanubile, and Nicole Novielli. How to ask for technical help? Evidence-based guidelines for writing questions on Stack Overflow. *Information and Software Technology*, 94:186–207, 2018.

[9] Jeffrey C. Carver, Natalia Juristo, Maria Teresa Baldassarre, and Sira Vegas. Replications of software engineering experiments. *Empirical Software Engineering*, 19(2):267–276, Apr 2014.

[10] Zhenpeng Chen, Yanbin Cao, Xuan Lu, Qiaozhu Mei, and Xuanzhe Liu. SEntiMoji: An Emoji-Powered Learning Approach for Sentiment Analysis in Software Engineering. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2019, page 841–852, New York, NY, USA, 2019. Association for Computing Machinery.

[11] Ward Cunningham. The wycash portfolio management system. *SIGPLAN OOPS Mess.*, 4(2):29–30, dec 1992.

[12] Fabio Q. B. da Silva, Marcos Suassuna, A. César C. França, Alicia M. Grubb, Tatiana B. Gouveia, Cleviton V. F. Monteiro, and Igor Ebrahim dos Santos. Replication of empirical studies in software engineering research: a systematic mapping study. *Empirical Software Engineering*, 19(3):501–557, Jun 2014.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[14] Joshua Aldrich Edbert, Sahrima Jannat Oishwee, Shubhashis Karmakar, Zadia Codabux, and Roberto Verdecchia. Exploring Technical Debt in Security Questions on

Stack Overflow. In *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2023, New Orleans, LA, USA, October 26-27, 2023*, pages 1–12. IEEE, 2023.

[15] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark, sep 2017. Association for Computational Linguistics.

[16] Eliakim Gama, Sávio Freire, Manoel Mendonça, Rodrigo O. Spínola, Matheus Paixao, and Mariela I. Cortés. Using Stack Overflow to Assess Technical Debt Identification on Software Projects. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*, SBES '20, page 730–739, New York, NY, USA, 2020. Association for Computing Machinery.

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

[18] Robbert Jongeling, Proshanta Sarkar, Subhajit Datta, and Alexander Serebrenik. On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering*, 22(5):2543–2584, Oct 2017.

[19] R. Kuhn, M. Raunak, and R. Kacker. Can reducing faults prevent vulnerabilities? *Computer*, 51(07):82–85, jul 2018.

[20] Zengyang Li, Paris Avgeriou, and Peng Liang. A systematic mapping study on technical debt and its management. *Journal of Systems and Software*, 101:193–220, 2015.

[21] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, Michele Lanza, and Rocco Oliveto. Sentiment Analysis for Software Engineering: How Far Can We Go? In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, pages 94–104, 2018.

[22] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113, 2014.

[23] Nicole Novielli, Fabio Calefato, Davide Dongiovanni, Daniela Girardi, and Filippo Lanubile. Can We Use SE-specific Sentiment Analysis Tools in a Cross-Platform Setting? In *Proceedings of the 17th International Conference on Mining Software Repositories*. ACM, jun 2020.

[24] Nicole Novielli, Fabio Calefato, Filippo Lanubile, and Alexander Serebrenik. Assessment of off-the-shelf se-specific sentiment analysis tools: An extended replication study. *Empirical Software Engineering*, 26(4):77, Jun 2021.

[25] Nicole Novielli, Daniela Girardi, and Filippo Lanubile. A Benchmark Study on Sentiment Analysis for Software Engineering Research. In *Proceedings of the 15th International Conference on Mining Software Repositories*, MSR '18, page 364–375, New York, NY, USA, 2018. Association for Computing Machinery.

[26] 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*, MSR 2014, page 348–351, New York, NY, USA, 2014. Association for Computing Machinery.

[27] Naveen Raman, Minxuan Cao, Yulia Tsvetkov, Christian Kästner, and Bogdan Vasilescu. Stress and burnout in open source: toward finding, understanding, and mitigating unhealthy interactions. In Gregg Rothermel and Doo-Hwan Bae, editors, *ICSE-NIER 2020: 42nd International Conference on Software Engineering, New Ideas and Emerging Results, Seoul, South Korea, 27 June - 19 July, 2020*, pages 57–60. ACM, 2020.

[28] Kalle Rindell and Johannes Holvitie. Security Risk Assessment and Management as Technical Debt. In *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pages 1–8, 2019.

[29] Klaus R. Scherer, Tanja Wranik, Janique Sangsue, Véronique Tran, and Ursula Scherer. Emotions in everyday life: probability of occurrence, risk factors, appraisal and reaction patterns. *Social Science Information*, 43(4):499–570, 2004.

[30] C.B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572, 1999.

[31] P Shaver, J Schwartz, D Kirson, and C O'Connor. Emotion knowledge: further exploration of a prototype approach. *J Pers Soc Psychol*, 52(6):1061–1086, jun 1987.

[32] Forrest J. Shull, Jeffrey C. Carver, Sira Vegas, and Natalia Juristo. The role of replications in empirical software engineering. *Empirical Software Engineering*, 13(2):211–218, Apr 2008.

[33] Gias Uddin, Md Abdullah Al Alamin, and Ajoy Das. An empirical study of deep learning sentiment detection tools for software engineering in cross-platform settings, 2023.

[34] Xin-Li Yang, David Lo, Xin Xia, Zhi-Yuan Wan, and Jian-Ling Sun. What Security Questions Do Developers Ask? A Large-Scale Study of Stack Overflow Posts. *Journal of Computer Science and Technology*, 31(5):910–924, Sep 2016.

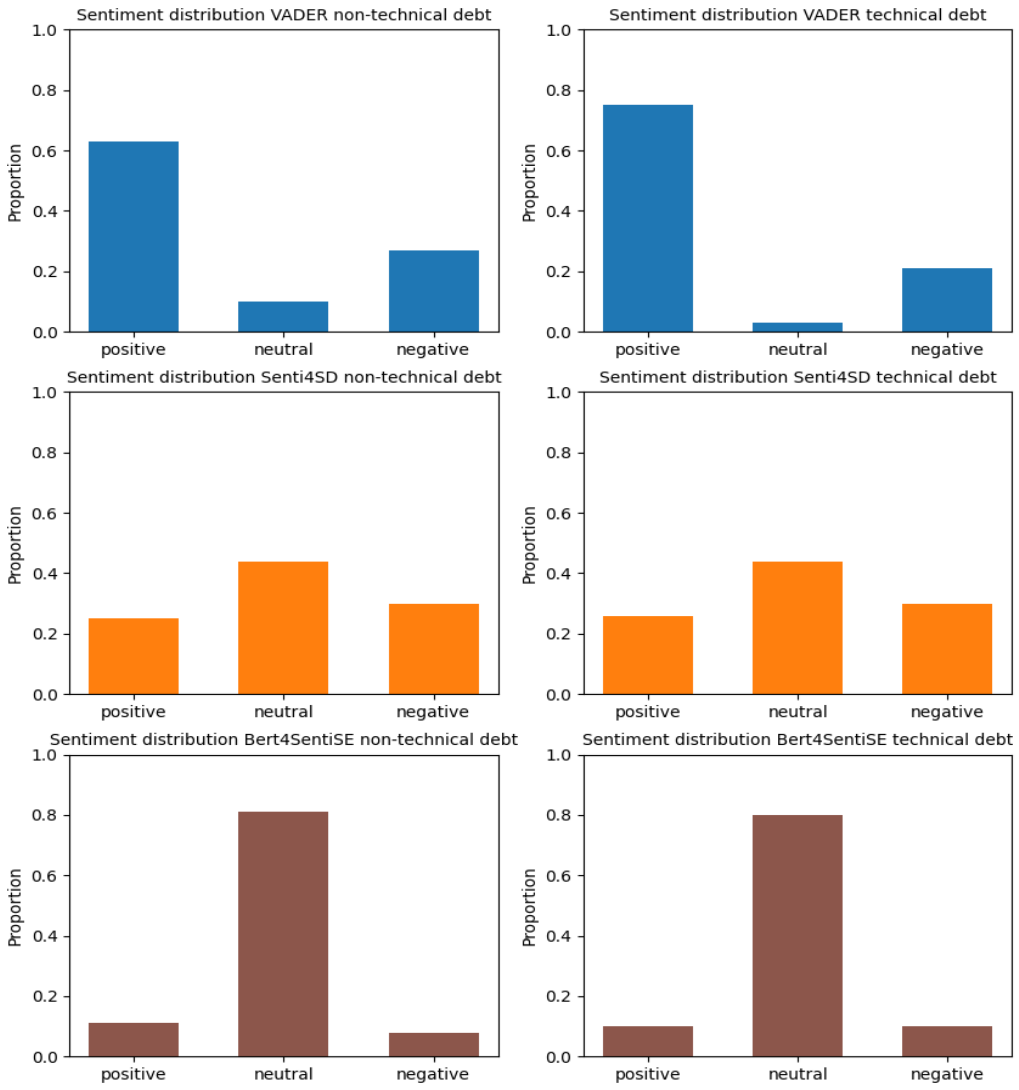

Fig. 1. Distribution of sentiment. The left column shows the distribution of sentiment on non-TD questions, the right column shows the distribution of sentiment on TD questions. The top row shows sentiment by VADER, the middle row sentiment by Senti4SD, and the bottom row sentiment by BERT4SentiSE.