Noname manuscript No.
(will be inserted by the editor)

On Negative Results when using Sentiment Analysis Tools for Software Engineering Research

Robbert Jongeling · Subhajit Datta · Alexander Serebrenik

the date of receipt and acceptance should be inserted later

Abstract Recent years have seen an increasing attention to social aspects of software engineering, including studies of emotions and sentiments experienced and expressed by the software developers. Most of these studies reuse existing sentiment analysis tools such as SentiStrength and NLTK. However, these tools have been trained on product reviews and movie reviews and, therefore, their results might not be applicable in the software engineering domain.

In this paper we study whether the sentiment analysis tools agree with the sentiment recognized by human evaluators (as reported in an earlier study) as well as with each other. Furthermore, we evaluate the impact of the choice of a sentiment analysis tool on software engineering studies by conducting a simple study of differences in issue resolution times for positive, negative and neutral texts. We repeat the study for seven datasets (issue trackers and Stack Overflow questions) and different sentiment analysis tools and observe that the disagreement between the tools can lead to contradictory conclusions. Finally, we perform two replications of previously published studies and observe that the results of those studies cannot be confirmed when a different sentiment analysis tool is used.

1 Introduction

Sentiment analysis is "the task of identifying positive and negative opinions, emotions, and evaluations" [66]. Since its inception sentiment analysis has been subject of an intensive research effort and has been successfully applied e.g., to assist users in their development by providing them with interesting and supportive content [27], predict the outcome of an election [61] or movie sales [36]. The spectrum of sentiment analysis techniques ranges from identifying polarity (positive or

Robbert Jongeling
Eindhoven University of Technology, The Netherlands E-mail: r.m.jongeling@student.tue.nl

Subhajit Datta
Singapore University of Technology and Design, Singapore E-mail: subhajit.datta@acm.org

Alexander Serebrenik
Eindhoven University of Technology, The Netherlands E-mail: a.serebrenik@tue.nl

negative) to a complex computational treatment of subjectivity, opinion and sentiment [42]. In particular, the research on sentiment polarity analysis has resulted in a number of mature and publicly available tools such as SentiStrength [57], Alchemy[1], Stanford NLP sentiment analyser [54] and NLTK [6].

Sentiment polarity analysis has been recently applied in the software engineering context to study commit comments in GitHub [24], GitHub discussions related to security [45], productivity in Jira issue resolution [40], activity of contributors in Gentoo [19], classification of user reviews for maintenance and evolution [44] and evolution of developers' sentiments in the openSUSE Factory [47]. It has also been suggested when assessing technical candidates on the social web [8]. Not surprisingly, all the aforementioned software engineering studies with the notable exception of the work by Panichella et al. [44], reuse the existing sentiment polarity tools, e.g., Pletea et al. [45] and Rousinopoulos et al. [47] use NLTK, while Garcia et al. [19], Guzman et al. [25,24], Novielli et al. [39] and Ortu et al. [40] opted for SentiStrength. While the reuse of the existing tools facilitated the application of the sentiment polarity analysis techniques in the software engineering domain, it also introduced a commonly recognized threat to validity of the results obtained: those tools have been trained on non-software engineering related texts such as movie reviews or product reviews and might misidentify (or fail to identify) polarity of a sentiment in a software engineering artefact such as a commit comment [24,45].

Therefore, in this paper we focus on sentiment polarity analysis [66] and investigate to what extent are the software engineering results obtained from sentiment analysis depend on the choice of the sentiment analysis tool. For the sake of simplicity, from here on, instead of "existing sentiment polarity analysis tools" we talk about the "sentiment analysis tools". Specifically, we aim at answering the following questions:

- *RQ1*: To what extent do different sentiment analysis tools agree with emotions of software developers?
- *RQ2*: To what extent do different sentiment analysis tools agree with each other?

We have observed disagreement between sentiment analysis tools and the emotions of software developers but also between different sentiment tools themselves. However, disagreement between the tools does not *a priori* mean that sentiment analysis tools might lead to contradictory results in software engineering studies making use of these tools. Thus, we ask

- *RQ3*: Do different sentiment analysis tools lead to contradictory results in a software engineering study?

We have observed that disagreement between the tools might lead to contradictory results in software engineering studies. Therefore, we finally conduct replication studies in order to understand:

- *RQ4*: How does the choice of a sentiment analysis tool affect validity of the previously published results?

The remainder of this paper is organized as follows. In Section 2 we study agreement between the tools and the results of manual labeling, and between the

[1] `http://www.alchemyapi.com/products/alchemylanguage/sentiment-analysis/`

tools themselves, i.e., *RQ1* and *RQ2*. In Section 3 we conduct a series of experiments based on the results of different sentiment analysis tools. We observe that conclusions one might derive using different tools diverge, casting doubt on their validity (*RQ3*). While our answer to *RQ3* indicates that the choice of a sentiment analysis tool *might* affect validity of software engineering results, in Section 4 we performing replication of two published studies answering *RQ4* and establishing that conclusions of the previously published work cannot be reproduced when a different sentiment analysis tool is used. Finally, in Section 5 we discuss related work and conclude in Section 6.

Source code used to obtain the results of this paper has been made available on GitHub[2].

2 Agreement between Sentiment Analysis Tools

In this section we address *RQ1* and *RQ2*, i.e., to what extent do different sentiment analysis tools agree with emotions of software developers and to what extent do different sentiment analysis tools agree with each other. To perform the evaluation we use the manually labeled emotions dataset [38].

2.1 Methodology

2.1.1 Sentiment Analysis Tools

We have considered four sentiment analysis tools SentiStrength, Alchemy, Stanford NLP sentiment analyser and NLTK. SentiStrength and NLTK have been used in earlier software engineering studies. Moreover, SentiStrength had the highest average accuracy among fifteen Twitter sentiment analysis tools [1]. The Stanford NLP parses the text into sentences and performs a more advanced grammatical analysis as opposed to a simpler bag of words model used in NLTK. Alchemy provides several text processing APIs, including a sentiment analysis API which promises to work on very short texts (e.g., tweets) as well as relatively long texts (e.g., news articles).

SentiStrength assigns an integer value between 1 and 5 for the positivity of a text, p and similarly, a value between -1 and -5 for the negativity, n. In order to map these scores to a document-level sentiment (positive, neutral or negative) for an entire text fragment, we follow the approach by Thelwall et al. [56] A text is considered positive when $p + n > 0$, negative when $p + n < 0$, and neutral if $p = -n$ and $p < 4$. Texts with a score of $p = -n$ and $p \geq 4$ are considered having an undetermined sentiment and are removed from the datasets.

Alchemy API returns for a text fragment a status, a language, a score and a type. The score is in the range $(-1, 1)$, the type is the sentiment of the text and is based on the score. For negative scores, the type is negative, conversely for positive scores, the type is positive. For a score of 0, the type is neutral. We ignore texts with status "ERROR" or a non-English language.

NLTK returns for each text a probability of it being negative, one of it being neutral and one of it being positive. If the probability score for neutral is greater

[2] `https://github.com/RobbertJongeling/ICSME2015ERA`

than 0.5, the text is considered neutral. Otherwise, it is considered to be the other sentiment with the highest probability [45]. To call NLTK, we use the API provided at text-processing.com.[3]

Stanford NLP breaks down the text into sentences and assigns each a sentiment score in the range $[0, 4]$, where 0 is very negative, 2 is neutral and 4 is very positive. We note that the tool may have difficulty breaking the text into sentences as comments sometimes include pieces of code or e.g. URLs. The tool does not provide a document-level score, to determine such a document-level sentiment we compute $-2 * \#0 - \#1 + \#3 + 2 * \#4$, where $\#0$ denotes the number of sentences with score 0, etc.. If this score is negative, neutral or positive, we consider the text to be negative, neutral or positive, respectively.

2.1.2 Manually-Labeled Software Engineering Data

As the "golden set" we use the data from a developer emotions study by Murgia et al. [38]. In this study, four evaluators manually labeled 392 comments with emotions "joy", "love", "surprise", "anger", "sadness" or "fear". Emotions "joy", "love" and "surprise" are taken as indicators of positive sentiments and "anger", "sadness" and "fear"—of negative sentiment.

We focus on consistently labeled comments. We consider the comment as positive if at least three evaluators have indicated a positive sentiment and no evaluators have indicated negative sentiments. Similarly, we consider the comment as negative if at least three evaluators have indicated a negative sentiment and no evaluators have indicated positive sentiments. We consider an evaluation neutral when an evaluator indicates no emotions. A text is then considered as neutral when three or more evaluators have evaluated it as neutral.

Using these rules we can conclude that 295 comments have been labeled consistently: 24 negative, 54 positive and 217 neutral. The remaining $392-24-54-217 = 97$ comments from the study Murgia et al. [38] have been labeled with contradictory labels e.g. "fear" by one evaluator and "surprise" by another.

2.1.3 Evaluation Metrics

Since more than 73% of the comments have been manually labeled as neutral, i.e., the dataset is unbalanced, traditional metrics such as accuracy might be misleading [4]: indeed, accuracy of the straw man sentiment analysis predicting "neutral" for any comment can be easily higher than of any of the four tools. Therefore, we use the Adjusted Rand Index (ARI) [48].

ARI measures the correspondence between two partitions of the same data: to answer the first research question we look for correspondence between the partition of the comments into positive, neutral and negative groups provided by the tool and the partition based on the manual labeling. Similarly, to answer the second research question we look for correspondence between partition of the comments into positive, neutral and negative groups provided by different tools. The expected value of ARI ranges for independent partitions is 0. The maximal value, obtained e.g., for identical partitions is 1, the closer the value of ARI to 1 the better the correspondence between the partitions.

[3] API docs for NLTK sentiment analysis: `http://text-processing.com/docs/sentiment.html`

Table 1: Agreement between the manual labeling, NLTK [6] and SentiStrength [57] on 295 comments

		Manual		
		neg	neu	pos
NLTK	neg	19	51	11
	neu	0	138	7
	pos	5	28	36

		SentiStrength		
		neg	neu	pos
NLTK	neg	17	39	25
	neu	15	96	34
	pos	6	20	43

2.2 Results

None of the 295 consistently labeled comments produce SentiStrength results with $p = -n$ and $p \geq 4$. Hence, no comments are excluded from the evaluation of SentiStrength. Five comments produce the "ERROR" status with Alchemy; those comments have been excluded from consideration, i.e., Alchemy has been evaluated on 290 comments.

When comparing the partitions induced by the sentiment analysis tools with the partition based on the manual labeling, the highest ARI has been obtained for NLTK (0.239), followed by SentiStrength (0.113), Stanford NLP (0.108) and Alchemy (0.079). When comparing the partitions induced by the sentiment analysis tools with each other we obtain low values of ARI: Alchemy API and NLTK have the highest ARI (0.104), followed by SentiStrength and NLTK (0.091).

Further details on the agreement between NLTK, SentiStrength and the manual labeling are shown in Table 1.

2.3 Discussion

Our results clearly indicate that the sentiment analysis tools do not agree with the manual labeling and neither do they agree with each other.

Given the disagreement between different sentiment analysis tools, we wonder whether combining the tools would result in a better agreement with the manual labeling. Thus, we have conducted a follow-up experiment: for every pair of tools we consider only comments on which the tools agree, and determine ARI with the manual labeling. For instance, for SentiStrength and NLTK we consider only $n = 156$ comments ($= 17+96+43$ on the main diagonal in Table 1 (right)). Results of the follow up study indicate that the best agreement with the manual labeling is achieved by combination of SentiStrength and NLTK (0.543, $n = 156$) followed by the combination of SentiStrength and Stanford NLP (0.318, $n = 121$), while the worst agreement has been observed for the combination of Alchemy and Stanford NLP (0.186, $n = 153$). In fact, the agreement of the SentiStrength/NLTK combination with the manual labeling is similar to the agreement with the manual labeling of the combination of all four tools we have considered (0.574); however, the number comments where the four tools agree ($n = 57$) is much smaller than where SentiStrength agrees with NLTK ($n = 156$). For the sake of completeness confusion matrices for the SentiStrength/NLTK combination and combination of the four tools are shown in Table 2.

Based on this evaluation we select SentiStrength and NLTK to address *RQ3*: these tools show the highest (albeit still low) degrees of correspondence with the

Table 2: Agreement between the manual labeling and tool combinations: SentiStrength/NLTK (left) and all four tools considered (right)

		Manual		
		neg	neu	pos
NLTK ∩ SS	neg	7	10	0
	neu	0	92	4
	pos	2	9	32

		Manual		
		neg	neu	pos
4 tools	neg	6	8	0
	neu	0	23	1
	pos	0	1	18

golden set. Moreover, these two sentiment analysis tools have one of the higher (albeit also low) agreements with each other. Furthermore, when discussing *RQ3* we also perform a separate analysis for texts upon which SentiStrength and NLTK agree as the follow-up experiment suggests that this combination has better agreement with the golden set.

2.4 Threats to Validity

As any empirical evaluation, the study presented in this section is subject to threats to validity. On top of the threats to validity inherent to the choice of the dataset used for evaluation and its construction [38] validity of our evaluation might have been affected by our decision to interpret emotions "joy", "love" and "surprise" as indicators of a positive sentiment and "anger", "sadness" and "fear"—as indicators of a negative sentiment. For instance, one might argue that not all surprises are positive. Therefore, we consider replication of this study on a manually labeled dataset as an important sent in the follow-up research.

Furthermore, the exact ways tools have been applied and the sentiment has been determined based on the tools' output, e.g., calculation of a document-level sentiment as $-2 * \#0 - \#1 + \#3 + 2 * \#4$ for Stanford NLP, might have affected validity of the conclusions.

3 Impact of the Choice of Sentiment Analysis Tool

In Section 2 we have seen that not only is the agreement of the sentiment analysis tools with the manual labeling limited, but also that different tools do not necessarily agree with each other. However, this disagreement does not necessarily mean that conclusions based on application of these tools in the software engineering domain are affected by the choice of the tool. Therefore, next we address *RQ3* and discuss a simple set-up of a study aiming at understanding differences in response times for positive, neutral and negative texts. While we do not aim at replicating an existing study, we note that similar questions have been considered in the literature [40, 19].

3.1 Methodology

We study whether differences can be observed between response times (issue resolution times or question answering times) for positive, neutral and negative texts.

We do not claim that the type of comment (positive, neutral or negative) is the main factor influencing response time: indeed, certain topics might be more popular than others and questions asked during the weekend might lead to higher resolution times. However, if different conclusions are derived for the same dataset when different sentiment analysis tools are used, then we can conclude that the disagreement between sentiment analysis tools affects validity of conclusions in the software engineering domain.

We repeat the study for seven different datasets: titles of issues of the Android issue tracker, descriptions of issues of the Android issue tracker, titles of issues of the Apache Software Foundation (ASF) issue tracker, descriptions of issues of the ASF issue tracker, descriptions of issues of the Gnome issue tracker, titles of the Gnome-related Stack Overflow questions and bodies of the Gnome-related Stack Overflow questions. As opposed to the Android dataset, Gnome issues do not have titles. For each dataset we determine the sentiment using NLTK and SentiStrength. Moreover, as suggested in Section 2.3 we repeat each study on the subset of texts where NLTK and SentiStrength agree.

3.1.1 Datasets

To ensure validity of our study we have analyzed texts from three independent datasets collected by other researchers (Android Issue Tracker, Gnome Issue Tracker, ASF Issue Tracker) and one dataset derived by us from a well-known public data source (Gnome-Related Stack Overflow Discussions). All datasets are publicly available for replication purposes[4].

Android *Issue Tracker.* A dataset of 20,169 issues from the Android issue tracker was part of the mining challenge of MSR 2012 [52]. Excluding issues without a closing date, as well as those with *bug_status* "duplicate", "spam" or "usererror", results in the dataset with 5,216 issues.

We analyze the sentiment of the issue titles and descriptions. Five issues have an *undetermined* description sentiment. We remove these issues from further analysis on the titles and the descriptions. To measure the response time, we calculate the time difference in seconds between the opening (*openedDate*) and closing time (*closedOn*) of an issue.

Gnome *Issue Tracker.* The Gnome project issue tracker dataset containing 431,863 issues was part of the 2009 MSR mining challenge[5]. Similarly to the Android dataset, we have looked only at issues with a value for field *bug_status* of `resolved`. In total 367,877 have been resolved. We analyze the sentiment of the short descriptions of the issues (*short_desc*) and calculate the time difference in seconds between the creation and closure of each issue. Recall that as opposed to the Android dataset, Gnome issues do not have titles.

[4] `https://github.com/RobbertJongeling/ICSME2015ERA`

[5] `http://msr.uwaterloo.ca/msr2009/challenge/msrchallengedata.html`

Gnome-*Related* Stack Overflow *Discussions.* We use the StackExchange online data explorer[6] to obtain all Stack Overflow posts created before May 20, 2015, tagged `gnome` and having an accepted answer. For all 410 collected posts, we calculate the time difference in seconds between the creation of the post and the creation of the accepted answer. Before applying a sentiment analysis tool we remove HTML formatting from the titles and bodies of posts. In the results, we refer to the `body` of a post as its description.

ASF *Issue Tracker.* We use a dataset containing data from the ASF issue tracking system Jira. This dataset was collected by Ortu et al. [40] and contains 701,002 issue reports. We analyze the sentiments of the titles and the descriptions of 95,667 issue reports that have a non-null resolved date, a *resolved* status and the resolution value being *Fixed*.

3.1.2 Statistical Analysis

To answer our research questions we need to compare distributions of response times corresponding to issues/questions bearing positive, neutral and negative sentiments. Traditionally, a comparison of multiple groups follows a two-step approach: first, a global null hypothesis is tested, then multiple comparisons are used to test sub-hypotheses pertaining to each pair of groups. The first step is commonly carried out by means of ANOVA or its non-parametric counterpart, the Kruskal-Wallis one-way analysis of variance by ranks. The second step uses the t-test or the rank-based Wilcoxon-Mann-Whitney test [65], with correction for multiple comparisons, e.g., Bonferroni correction [14,51]. Unfortunately, the global test null hypothesis may be rejected while none of the sub-hypotheses are rejected, or vice versa [17]. Moreover, simulation studies suggest that the Wilcoxon-Mann-Whitney test is not robust to unequal population variances, especially in the case of unequal sample sizes [7,67]. Therefore, one-step approaches are preferred: these should produce confidence intervals which always lead to the same test decisions as the multiple comparisons. We use the $\widetilde{\mathbf{T}}$-procedure [30] for Tukey-type contrasts [60], the probit transformation and the traditional 5% family error rate (cf. [62,64]).

The results of the $\widetilde{\mathbf{T}}$-procedure are series of probability estimates $p(a, b)$ with the corresponding p-values, where a and b are selected from "positive", "neutral" or "negative". The probability estimate $p(a, b)$ is interpreted as follows: if the corresponding p-value exceeds 5% then no evidence has been found for difference in response times corresponding to categories a and b. If, however, the corresponding p-value does not exceed 5% and $p(a, b) > 0.5$ then response times in category b tends to be larger than those in category a. Finally, if the corresponding p-value does not exceed 5% and $p(a, b) < 0.5$ then response times in category a tends to be larger than those in category b.

We opt for comparison of distributions rather than a more elaborate statistical modeling (cf. [40]) since it allows for an easy comparison of the results obtained for different tools.

[6] `http://data.stackexchange.com/`

Table 3: Comparison of NLTK and SentiStrength. Thresholds for statistical significance: 0.05 (*), 0.01 (**), 0.001 (***).

	NLTK neg-neu-pos	SentiStrength neg-neu-pos	NLTK ∩ SentiStrength neg-neu-pos
		Android	
title	1,230-3,588-398 ∅	1,417-3,415-384 ∅	396-2,381-36 ∅
descr	2,690-1,657-869 neu > neg*** **neu > pos****	1,684-2,435-1,182[7] **neu > pos**** neg > pos***	893-712-299 neu > neg* **neu > pos*** neg > pos*
		Gnome	
descr	54,032-291,906-20,380 **neg > neu*** **pos > neu*** pos > neg***	58,585-293,226-14,507 **neg > neu*** **pos > neu*** neg > pos***	16,829-24,2780-1,785 **neg > neu*** **pos > neu***
		Stack Overflow	
title	84-285-41 ∅	53-330-27 ∅	16-240-8 ∅
descr	249-71-90 ∅	90-183-137 neg > pos*	62-35-42 ∅
		ASF	
title	19,367-67,948-8,348[8] pos > neg*	24,141-62,016-9,510 neg > neu** pos > neu*** pos > neg***	6,450-44,818-1,106 pos > neu**
descr[9]	30,339-42,540-13,129[10] neg > neu*** **pos > neu***	29,021-41,043-15,971[11] **pos > neu*** pos > neg***	10,989-20,940-3,814 neg > neu*** **pos > neu*** pos > neg***

3.2 Results

Results of our study are summarized in Table 3. For the sake of readability the relations found are aligned horizontally. Relations found for NLTK, SentiStrength *and* the combination of the tools are typeset in boldface. We also report the number of issues/questions recognized as negative, neutral or positive.

We observe that NLTK and SentiStrength agree only on one relation for the Android, i.e., that issues with the neutral sentiment tend to be resolved more slowly than issues formulated in a more positive way. We also observe that for Gnome and ASF the tools agree that the issues with the neutral sentiment are resolved faster than issues with the positive sentiment, i.e., the results for Gnome and ASF are opposite from those for Android.

Further inspection reveals that differences between NLTK and SentiStrength led to relations "neu > neg" and "neg > pos" to be discovered in Android issue

[7] Sentiment of 5 issues was "undetermined".

[8] The tool reported an error for 4 issues

[9] 9,620 empty descriptions where not included in this analysis

[10] The tool reported an error for 39 issues

[11] Sentiment of 12 issues was "undetermined".

descriptions only by one of the tools and not by the other. In the same way, "pos > neg" on the ASF descriptions data can be found only by SentiStrength. It is also surprising that while "pos > neg" has been found for the ASF titles data both by NLTK and by SentiStrength, it cannot be found when one restricts the attention to the issues where the tools agree. Finally, contradictory results have been obtained for Gnome issue descriptions: while the NLTK-based analysis suggests that the positive issues are resolved more slowly than the negative ones, the SentiStrength-based analysis suggests the opposite.

3.3 Discussion

Our results suggest the choice of the sentiment analysis tool affects the conclusions one might derive when analysing differences in the response times, casting doubt on the validity of those conclusions. We conjecture that the same might be observed for any kind of software engineering studies dependent on off-the-shelf sentiment analysis tools. A more careful sentiment analysis for software engineering texts is therefore needed: e.g., one might consider training more general purpose machine learning tools such as Weka [26] or RapidMiner[12] on software engineering data.

A similar approach has been recently taken by Panichella et al. [44] that have used Weka to train a Naive Bayes classifier on 2090 App Store and Google Play review sentences. Indeed, both dependency of sentiment analysis tools on the domain [18] and the need for text-analysis tools specifically targeting texts related to software engineering [28] have been recognized in the past.

4 Implications on Earlier Studies

In this section we consider *RQ4*: while the preceding discussion indicates that the choice of a sentiment analysis tool *might* affect validity of software engineering results, in this section by performing replication studies [53] we investigate whether this is indeed the case for two published examples. Since our goal is to understand whether the effects observed in the earlier studies hold when a different sentiment analysis tool is used, we opt for *dependent or similar* replications [53].

4.1 Replicated studies

We have chosen to replicate two previous studies conducted as part of the 2014 MSR mining challenge: both studies use the same dataset of 90 GitHub projects [22]. The dataset includes information from the top-10 starred repositories in the most popular programming languages and is not representative for GitHub as a whole[13].

The first paper we have chosen to replicate is the one by Pletea et al. [45]. In this paper the authors apply NLTK to GitHub comments and discussions, and conclude that security-related discussions on GitHub contain more negative emotions than non-security related discussions. Taking the blame, the third author of the current manuscript has also co-authored the work by Pletea et al. [45].

[12] `https://rapidminer.com/solutions/sentiment-analysis/`

[13] `http://ghtorrent.org/msr14.html`

The second paper we have chosen to replicate is the one by Guzman et al. [24]. The authors apply SentiStrength to analyze the sentiment of GitHub commit comments and conclude that comments written on Mondays tend to contain a more negative sentiment than comments written on other days.

4.2 Replication approach

We replicate the studies exactly with one notable deviation from the original work: we apply a different sentiment analysis tool to each study. Since the original study of Pletea et al.uses NLTK, we apply SentiStrength in the replication; since Guzman et al.use SentiStrength, we apply NLTK. We hypothesize that we might get different, statistically significant, results in these studies when using a different sentiment analysis tool.

4.2.1 Pletea et al.

Pletea et al.distinguish between *comments* and *discussions*, collections of comments pertaining to an individual commit or pull request. Furthermore, the authors distinguish between security-related and non-security related comments/discussions, resulting in eight different categories of texts. The original study has found that for commits comments, commit discussions, pull request comments and pull request discussions, the negativity for security related texts is higher that for other texts.

In our replication of this study we present a summary of the distribution of the sentiments for commits and pull requests, recreating Tables 2 and 3 from the original study. In order to do this, we also need to distinguish security-related texts and other texts, i.e., we replicate Table 1 from the paper. Furthermore the original paper presents a manual case study of 30 discussions to evaluate the performance of the sentiment analysis tool. We extend their table to include the results obtained by SentiStrength.

4.2.2 Guzman et al.

In this study, the authors have focused on commit comments and studied differences between the sentiment of commit comments written at different days of week and times of day, belonging to projects in different programming languages, created by teams distributed over different continents and "starred", i.e., approved, by different number of GitHub users.

We replicate the studies pertaining to differences between comments based on day and time of their creation and programming language of the project. We do not replicate the study related to the geographic distribution of the authors because the mapping of developers to continents has been manually made by Guzman et al.and was not present in the original dataset.

4.3 Replication results

Here we present the results of replicating both studies.

Table 4: Identification of security-related comments and discussions results

Type			Comments	Discussions
Commits	Pletea et al. [45]	Security	2689 (4.43%)	1809 (9.84%)
		Total	60658	18380
	Current study	Security	2567 (4.23%)	1761 (9.58%)
		Total	60658	18378
Pull Requests	Pletea et al. [45]	Security	1932 (3.51%)	1158 (12.06%)
		Total	54892	9602
	Current study	Security	1790 (3.25%)	1105 (11.51%)
		Total	54892	9601

4.3.1 Pletea et al.

We start the replication by creating Table 4, which corresponds to Table 1 from the paper by Pletea et al.. We have rerun the division using the keyword list as included in the original paper. We have found slightly different numbers of comments and discussions in each category. Most notably we find 142 less security-related comments in pull requests. However, the percentages of security and non-security related comments and discussions are similar.

To verify that the minor differences between the datasets of Pletea et al. [45] and of the current study shown in Table 4 do not influence the results, we have first applied *the same tool* as in the original study, i.e.,NLTK, to classify the sentiments in commit comments. The sentiment proportions per category in both studies are shown in Table 5; both for security and for non-security comments the p value of the χ^2 test exceeds 0.99, i.e., the differences shown in Table 4 indeed do not influence distribution of the sentiments.

Next we apply SentiStrength to analyze the sentiment of comments and discussions. Tables 6 and 7 present the results Tables 2 and 3 of the original paper, respectively, and extend them by including results of SentiStrength. The tables clearly show that the NLTK-based conclusion that security related texts are more negative is not supported when classifying the texts with SentiStrength. According to SentiStrength neutral is the predominant classification in all types of texts.

Finally, in Table 4 Pletea et al.consider thirty discussions and compare evaluation of the security relevance and sentiment as determined by the tools with the decisions performed by the human evaluator. The discussions have been selected based on the number of security keywords found: ten discussions labeled as “high” have been randomly selected from the top 10% discussions with the highest number of security keywords found, “middle” from the middle 10% and “low” from

Table 5: Commit comment sentiments by Pletea et al.and this study, both using NLTK

Type		Negative	Neutral	Positive
Pletea et al. [45], NLTK	Security	55.59%	23.42%	20.97%
	Rest	46.94%	26.58%	26.47%
Current study, NLTK	Security	55.94%	23.02%	20.96%
	Rest	46.86%	26.59%	26.51%

Table 6: Commits sentiment analysis statistics. The largest group per experiment is typeset in boldface.

Type			Negative	Neutral	Positive
Discussions	Pletea et al. [45]	Security	**72.52%**	10.88%	16.58%
	NLTK	Rest	**52.28%**	20.37%	25.33%
	Current study	Security	29.59%	**42.36%**	27.09%
	SentiStrength	Rest	24.09%	**44.05%**	31.77%
Comments	Pletea et al. [45]	Security	**55.59%**	23.42%	20.97%
	NLTK	Rest	**46.94%**	26.58%	26.47%
	Current study	Security	31.55%	**46.71%**	21.74%
	SentiStrength	Rest	22.29%	**51.04%**	26.66%

Table 7: Pull Requests sentiment analysis statistics. The largest group per experiment is typeset in boldface.

Type			Negative	Neutral	Positive
Discussions	Pletea et al. [45]	Security	**81.00%**	5.52%	13.47%
	NLTK	Rest	**69.58%**	11.98%	18.42%
	Current study	Security	29.68%	**45.70%**	23.71%
	SentiStrength	Rest	24.06%	**51.33%**	24.54%
Comments	Pletea et al. [45]	Security	**59.83%**	19.09%	21.06%
	NLTK	Rest	**50.16%**	26.12%	23.70%
	Current study	Security	24.58%	**51.28%**	21.08%
	SentiStrength	Rest	18.10 %	**64.12%**	18.69%

the bottom 10% of all security-related discussions. Table 8 extends Table 4 [45] by adding a column with the results of SentiStrength.

By inspecting Table 8 we observe that NLTK agrees with the human evaluator in 14 cases out of 30; SentiStrength—in 13 cases out of 30 but the tools agree with each other only in 9 cases. Based on the comparison with the manual evaluation Pleteaet al.observe that while security related discussions may not necessarily be more negative than other discussions, they appear to be more emotional. This observation is not supported by SentiStrength that classifies 17 out of 30 discussions as neutral.

4.3.2 Guzman et al.

We classified all commit comments in the MSR 2014 challenge dataset [22] using NLTK. Of the total of 60658 comments, 60634 could be classified (NLTK reported an error for the remaining 24 comments).

In the original paper by Guzman et al. [24] the authors claim to have analyzed 60425 commit comments, on the one hand, to have focused on comments of all projects having more than 200 comments, on the other. However, when replicating this study and considering comments of projects having more than 200 comments we have obtained merely 50133 comments, more then ten thousand comments less than in the original study. Therefore, to be as close as possible to the original study we have decided to include *all* commit comments in the dataset which produced 233 comments more than in the original study.

Guzmanet al.start by considering six projects with the highest number of commit comments: Jquery, Rails, CraftBukkit, Diaspora, MaNGOS and TrinityCore.

Table 8: Case study results (sentiments labeled on a 5-star scale).

Sec. relevance	Discussion (Commit ID)	# sec. keywords	Sec. relevance (human)	NLTK neutral (%)	NLTK negative (%)	NLTK positive (%)	NLTK result	SentiStrength result	Sentiment (human)
High	535033	6	Yes	16.5	42.9	57.0	pos	neutral	neg(*)
	256855	4	Yes	17.1	84.2	15.7	neg	neutral	neg(*)
	455971	6	Yes	19.1	84.3	15.6	neg	neutral	neutral
	131473	5	Yes	21.4	45.8	54.2	pos	neg	neg(*****)
	253685	4	No	20.4	59.1	40.8	neg	neutral	pos(*)
	370765	5	Yes	20.0	65.0	34.9	neg	neutral	pos(***)
	59082	4	No	19.8	76.4	23.5	neg	neutral	neg(*)
	157981	11	Yes	23.9	58.8	41.1	neg	neutral	neg(***)
	391963	9	Yes	16.7	71.9	28.0	neg	neutral	pos(****)
	272987	4	Yes	22.4	41.6	58.3	pos	pos	neg(*)
Medium	15128	1	No	20.6	71.3	28.6	neg	neutral	neutral
	396099	1	No	18.8	74.0	26.0	neg	neg	neg(****)
	132779	1	No	30.6	76.4	23.5	neg	pos	neutral
	295686	1	No	23.9	70.7	29.3	neg	neutral	pos(*)
	541007	1	Partial	37.7	71.7	28.2	neg	neg	neg(*)
	199287	1	Partial	18.9	76.4	23.5	neg	neutral	neg(*)
	461318	1	Yes	15.0	75.0	24.9	neg	neutral	neg(*)
	509384	1	Partial	33.4	67.3	32.7	neg	neutral	neutral
	338681	1	No	29.9	75.5	24.4	neg	pos	neg(*)
	511734	1	No	17.6	79.4	20.5	neg	pos	pos(***)
Low	364215	1	No	41.4	44.1	55.8	pos	neg	neg(*)
	274571	1	Partial	30.1	46.5	53.4	pos	pos	neg(**)
	47639	1	Yes	19.3	38.6	61.3	pos	neutral	pos(*****)
	277765	1	No	27.0	45.2	54.7	pos	pos	pos(*)
	6491	1	No	37.6	29.6	70.4	pos	neutral	neutral
	130367	1	No	15.4	43.6	56.3	pos	pos	pos(*)
	189623	1	No	57.9	35.8	64.1	neutral	neutral	pos(***)
	41379	1	Partial	30.9	26.1	73.8	pos	pos	pos(***)
	456580	1	No	26.6	46.6	53.3	pos	pos	pos(***)
	52122	1	No	17.6	46.3	53.6	pos	neutral	pos(*****)

The authors present two charts to show the average sentiment score in those six projects and the proportions of negative, neutral and positive sentiments in commit comments. Figures 2 and 4 show the results of our replications of these charts. As shown in Figure 2, the average emotion score is more negative than in the original study for each project, which is shown in Figure 1. In Figure 4 a larger proportion of negative commit comments is shown than in the original paper, which is reproduced in Figure 3.

Tables 9–11 contain the results from replicating the experiments done in the study by Guzman et al.using NLTK instead of SentiStrength.

In contrast to SentiStrength, the tool we applied outputs scores between 0 and 1 for negative, neutral and positive to indicate the probability of each sentiment. In the original paper, the SentiStrength scores are mapped to an integer in the range $[-5, -1)$ for negative texts, 0 for neutral texts and in the range $(1, 5]$ for positive texts. In addition, negative scores were multiplied by 1.5 to account for the less frequent occurrence of negativity in human texts. Therefore, we apply

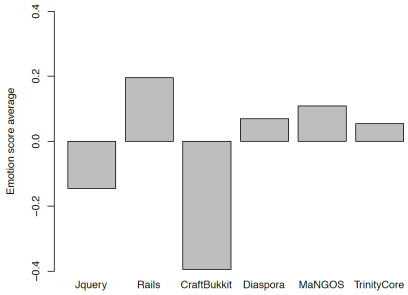

Fig. 1: Emotion score average per project, using SentiStrength [24]

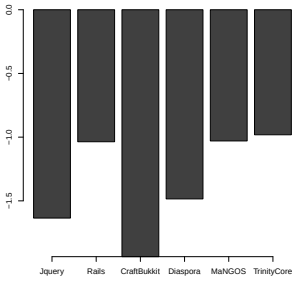

Fig. 2: Emotion score average per project, using NLTK (replication)

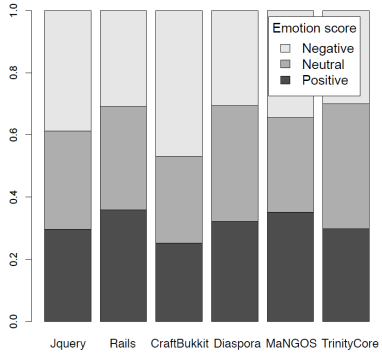

Fig. 3: Proportion of positive, neutral and negative commit comments per project, using SentiStrength [24]

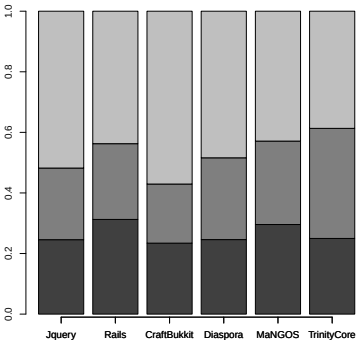

Fig. 4: Proportion of positive, neutral and negative commit comments per project, using NLTK (replication)

a transformation to create numbers in the same range according to the following formula:

$$sentiment\_score = \begin{cases} (((neg - 0.5) * (-6)) - 2) * 1.5 & \text{if } neg \\ 0 & \text{if } neutral \\ ((pos - 0.5) * 6) + 2 & \text{if } pos \end{cases}$$

The formula maps numbers from the range given by NLTK to the range used by SentiStrength as well as multiplies negative comments by 1.5, as done in the study by Guzman et al..

We stress that we do not compare the sentiment values obtained using NLTK with those obtained using SentiStrength. Rather we compare sentiment values obtained for different groups of comments using the same tool, and then observe (dis)agreement between the conclusions made based on different tools.

Table 9: Emotion score average grouped by programming language.

Language	Guzman et al. [24] SentiStrength			Current study NLTK		
	Comm.	**Mean**	**Stand. Dev.**	**Comm.**	**Mean**	**Stand. Dev.**
C	6257	0.023	1.716	6271	-1.833	3.0945
C++	16930	0.017	1.725	16978	-1.017	2.956
Java	4713	-0.144	1.736	4714	-1.755	3.1056
Python	2128	-0.018	1.711	2134	-1.635	3.079
Ruby	15257	0.002	1.714	15353	-1.243	3.117

Table 10: Emotion score average grouped by weekday.

Weekday	Guzman et al. [24] SentiStrength			Current study NLTK		
	Comm.	**Mean**	**Stand. Dev.**	**Comm.**	**Mean**	**Stand. Dev.**
Monday	9517	-0.043	1.732	9532	-1.316	3.047
Tuesday	9319	0.005	1.712	9385	-1.342	3.079
Wednesday	9730	0.008	1.716	9744	-1.370	3.100
Thursday	9538	0.001	1.728	9560	-1.358	3.073
Friday	9076	-0.016	1.739	9153	-1.347	3.082
Saturday	6701	-0.027	1.688	6720	-1.323	3.066
Sunday	6544	0.022	1.717	6540	-1.382	3.081

Table 11: Emotion score average grouped by time of the day.

Time of Day	Guzman et al. [24] SentiStrength			Current study NLTK		
	Comm.	**Mean**	**Stand. Dev.**	**Comm.**	**Mean**	**Stand. Dev.**
morning	12714	0.001	1.730	12744	-1.397	3.062
afternoon	19809	0.004	1.717	19852	-1.327	3.076
evening	16584	-0.023	1.721	16629	-1.323	3.085
night	11318	-0.016	1.713	11409	-1.369	3.077

Table 9 shows a lower average emotion score for the C programming language than for Java, indicating that we cannot derive the conclusion of Guzmanet al.that comments from projects written in Java are more negative that comments from projects written in C. Similarly, we cannot replicate the conclusion that comments on Monday were more negative than comments on the other days. In Table 10 the mean emotion score for Monday is the least negative. Finally, Table 11 shows that comments made in the afternoon are slightly more negative than comments in the evening, in contrast to the conclusions based on SentiStrength.

The original paper has also studied correlation between a project average sentiment score and its number of stars. Both the original work and our replication found no such correlation: the original study did not report the exact statistical techniques used, for the replication study Spearman $\rho \sim 0.07$ and not statistically significant, $p \sim 0.58$. When considering only the positive comments per project the original study reported a positive weak correlation ($\rho \sim 0.316$), while we do not find correlation at all: $\rho \sim 0.06$ and not statistically significant ($p \sim 0.62$).

4.4 Discussion

For both replication studies we have observed that the results presented in the original papers cannot be replicated when a different sentiment analysis tool is used. Validity of the conclusions of those papers should therefore be questioned and ideally reassessed when (or if) a sentiment analysis tool will become available specifically targeting software engineering domain.

4.5 Threats to validity

As any empirical study the current replications are subject to threats to validity. Since we have tried to follow the methodology presented in the papers being replicated as closely as possible, we have also inherited some of the threats to validity of those papers, e.g., that the dataset under consideration is not representative for GitHub as a whole. Furthermore, we had to convert the NLTK scores to the $[-5, 5]$ scale and this conversion might have introduced additional threats to validity. Finally, we are aware that discussing means and standard deviations as done in Section 4.3.2 might not be sound from the statistical point of view: this is why a more advanced statistical technique has been used in Section 3. However, to support the comparative aspects of replication in Section 4.3.2 we present the results exactly in the same way as in the original work [24].

5 Related Work

This paper builds on our previous work [29]. The current submission extends it by reporting on a follow-up study (Section 2.3), replication of two recent studies (Section 4) as well presenting a more elaborate discussion of the related work below.

5.1 Sentiment analysis in large text corpora

As announced in the *Manifesto for Agile Software Development* [5], the centrality of developer interaction in large scale software development has come to be increasingly recognized in recent times [11], [49]. Today, software development is influenced in myriad ways by how developers talk, and what they talk about. With distributed teams developing and maintaining many software systems today [9], developer interaction is facilitated by collaborative development environments that capture details of discussion around development activities [10]. Mining such data offers an interesting opportunity to examine implications of the sentiments reflected in developer comments.

Since its inception, sentiment analysis has become a popular approach towards classifying text documents by the predominant sentiment expressed in them [43]. As people increasingly express themselves freely in online media such as the microblogging site Twitter, or in product reviews on Web marketplaces such as Amazon, rich corpora of text are available for sentiment analysis. Davidov et al., have

suggested a semi-supervised approach for recognizing sarcastic sentences in Twitter and Amazon [12]. As sentiments are inherently nuanced, a major challenge in sentiment analysis is to discern the contextual meaning of words. Pak and Patrick suggest an automated and language independent method for disambiguating adjectives in Twitter data [41] and Agarwal et al., have proposed an approach to correctly identify the polarity of tweets [2]. Mohammad, Kiritchenko, and Xiaodan report the utility of using support vector machine (SVM) base classifiers while analyzing sentiments in tweets [37]. Online question and answer forums such as Yahoo! Answers are also helpful sources for sentiment mining data [31].

5.2 Sentiment analysis application in software engineering

The burgeoning field of tools, methodologies, and results around sentiment analysis have also impacted how we examine developer discussion. Goul et al.examine how requirements can be extracted from sentiment analysis of app store reviews [21]. The authors conclude that while sentiment analysis can facilitate requirements engineering, in some cases algorithmic analysis of reviews can be problematic [21]. User reviews of a software system in operation can offer insights into the quality of the system. However given the unstructured nature of review comments, it is often hard to reach a clear understanding of how well a system is functioning. A key challenge comes from "... different sentiment of the same sentence in different environment". To work around this problem, Leopairote et al.propose a methodology that combines lists of positive and negative sentiment words with rule based classification [32]. Mailing lists often characterize large, open source software systems as different stakeholders discuss their expectations as well as disappointments from the system. Analyzing the sentiment of such discussions can be an important step towards a deeper understanding of the corresponding ecosystem. Tourani et al.seek to identify distress or happiness in a development team by analyzing sentiments in Apache mailing lists [59]. The study concludes that developer and user mailing lists carry similar sentiments, though differently focused; and automatic sentiment analysis techniques need to be tuned specifically to the software engineering context [39].

As mentioned earlier, developer interaction data captured by collaborative development environments are fertile grounds for analyzing sentiments. There are recent trends around designing emotion aware environments that employ sentiment analysis and other techniques to discern and visualize health of a development team in real time [63]. Latest studies have also explored the symbiotic relationship between collaborative software engineering and different kinds of task based emotions [13].

5.3 Sentiment analysis tools

As already mentioned in the introduction, application of sentiment analysis tools to software engineering texts has been studied in a series of recent publications [19, 24, 25, 39, 40, 44, 45, 47]

With the notable exception of the work of Panichella et al. [44] that trained their own classifier on manually labeled software engineering data, all other works

have reused the existing sentiment analysis tools. As such reuse of those tools introduced a commonly recognized threat to validity of the results obtained: those tools have been trained on non-software engineering related texts such as movie reviews or product reviews and might misidentify (or fail to identify) polarity of a sentiment in a software engineering artefact such as a commit comment [24, 45].

In our previous work [29] and in the current submission we perform a series of quantitative analyses aiming at evaluation whether the choice of the sentiment analysis tool can affect the validity of the software engineering results. A complementary approach to evaluating the applicability of sentiment analysis tools to software engineering data has been followed by Novielli et al.. [39] that performed a qualitative analysis of Stack Overflow posts and compared the results of SentiStrength with those obtained by manual evaluation.

Beyond the discussion of sentiment analysis tools observations similar to those we made have been made in the past for software metric calculators [3] and code smell detection tools [15]. Similarly to our findings, disagreement between the tools was observed.

5.4 Replications and negative results

This paper builds on our previous work [29]. The current submission extends it by reporting on replication of two recent studies (Section 4). There is an enduring concern about the lack of replication studies in empirical software engineering: "Replication is not supported, industrial cases are rare ... In order to help the discipline mature, we think that more systematic empirical evaluation is needed." [58]. The challenges around replication studies in empirical software engineering have been identified in [35]. de Magalh et al.analyzed 36 papers reporting empirical and non-empirical studies related to replications in software engineering and concluded that not only do we need to replicate more studies in software engineering, expansion of "specific conceptual underpinnings, definitions, and process considering the particularities" are also needed [34]. Recent studies have begun to address this replication gap [50, 23].

One of the most important benefits of replication studies center around the possibility of arriving at negative results. Although negative results have been widely reported and regarded in different fields of computing since many years [46, 16], its importance is being reiterated in recent years [20]. By carefully and objectively examining what went wrong in the quest for expected outcome, the state-of-art and practice can be enhanced [33, 55]. We believe the results reported in this paper can aid such enhancement.

6 Conclusions

In this paper we have studied the impact of the choice of a sentiment analysis tool when conducting software engineering studies. We have observed that not only do the tools considered not agree with the manual labeling, but also they do not agree with each other, that this disagreement can lead to contradictory conclusions and that previously published results cannot be replicated when different sentiment analysis tools are used.

Our results suggest a need for sentiment analysis tools specially targeting the software engineering domain. Moreover, going beyond the specifics of the sentiment analysis domain, we would like to encourage the researchers to reuse ideas rather than tools.

Acknowledgements

We are very grateful to Alessandro Murgia and Marco Ortu for making their datasets available for our study, and to Bogdan Vasilescu and reviewers of ICSME 2015 for providing feedback on the preliminary version of this manuscript.

References

1. Abbasi, A., Hassan, A., Dhar, M.: Benchmarking Twitter sentiment analysis tools. In: International Conference on Language Resources and Evaluation, pp. 823–829. ELRA, Reykjavik, Iceland (2014)
2. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment Analysis of Twitter Data. In: Proceedings of the Workshop on Languages in Social Media, LSM '11, pp. 30–38. Association for Computational Linguistics, Stroudsburg, PA, USA (2011). URL http://dl.acm.org/citation.cfm?id=2021109.2021114
3. Barkmann, H., Lincke, R., Löwe, W.: Quantitative evaluation of software quality metrics in open-source projects. In: IEEE International Workshop on Quantitative Evaluation of large-scale Systems and Technologies, pp. 1067–1072 (2009)
4. Batista, G.E.A.P.A., Carvalho, A.C.P.L.F., Monard, M.C.: Applying one-sided selection to unbalanced datasets. In: Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence, pp. 315–325. Springer-Verlag, London, UK, UK (2000)
5. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for agile software development. http://agilemanifesto.org/principles.html Last accessed: October 14, 2015 (2001)
6. Bird, S., Loper, E., Klein, E.: Natural Language Processing with Python. O'Reilly Media Inc. (2009)
7. Brunner, E., Munzel, U.: The Nonparametric Behrens-Fisher Problem: Asymptotic Theory and a Small-Sample Approximation. Biometrical Journal **42**(1), 17–25 (2000)
8. Capiluppi, A., Serebrenik, A., Singer, L.: Assessing technical candidates on the social web. Software, IEEE **30**(1), 45–51 (2013). DOI 10.1109/MS.2012.169
9. Cataldo, M., Herbsleb, J.D.: Communication networks in geographically distributed software development. In: Proceedings of the 2008 ACM conference on Computer supported cooperative work, CSCW '08, p. 579588. ACM, New York, NY, USA (2008). DOI 10.1145/1460563.1460654. URL http://doi.acm.org/10.1145/1460563.1460654
10. Costa, J.M., Cataldo, M., de Souza, C.R.: The scale and evolution of coordination needs in large-scale distributed projects: implications for the future generation of collaborative tools. In: Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11, p. 31513160. ACM, New York, NY, USA (2011). DOI 10.1145/1978942.1979409. URL http://doi.acm.org/10.1145/1978942.1979409
11. Datta, S., Sindhgatta, R., Sengupta, B.: Talk versus work: characteristics of developer collaboration on the jazz platform. In: Proceedings of the ACM international conference on Object oriented programming systems languages and applications, OOPSLA '12, pp. 655–668. ACM, New York, NY, USA (2012). DOI 10.1145/2384616.2384664. URL http://doi.acm.org/10.1145/2384616.2384664
12. Davidov, D., Tsur, O., Rappoport, A.: Semi-supervised Recognition of Sarcastic Sentences in Twitter and Amazon. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning, CoNLL '10, pp. 107–116. Association for Computational Linguistics, Stroudsburg, PA, USA (2010). URL http://dl.acm.org/citation.cfm?id=1870568.1870582

13. Dewan, P.: Towards Emotion-Based Collaborative Software Engineering. In: 2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), pp. 109–112 (2015). DOI 10.1109/CHASE.2015.32
14. Dunn, O.J.: Multiple comparisons among means. Journal of the American Statistical Association **56**(293), 52–64 (1961)
15. Fontana, F.A., Mariani, E., Morniroli, A., Sormani, R., Tonello, A.: An experience report on using code smells detection tools. In: ICST Workshops, pp. 450–457. IEEE (2011)
16. Fuhr, N., Muller, P.: Probabilistic search term weighting - some negative results. In: Proceedings of the 10th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '87, pp. 13–18. ACM, New York, NY, USA (1987). DOI 10.1145/42005.42007. URL `http://doi.acm.org.library.sutd.edu.sg:2048/10.1145/42005.42007`
17. Gabriel, K.R.: Simultaneous test procedures—some theory of multiple comparisons. ANN MATH STAT **40**(1), 224–250 (1969)
18. Gamon, M., Aue, A., Corston-Oliver, S., Ringger, E.: Pulse: Mining customer opinions from free text. In: Proceedings of the 6th International Conference on Advances in Intelligent Data Analysis, IDA'05, pp. 121–132. Springer-Verlag, Berlin, Heidelberg (2005). DOI 10.1007/11552253_12. URL `http://dx.doi.org/10.1007/11552253_12`
19. Garcia, D., Zanetti, M.S., Schweitzer, F.: The role of emotions in contributors activity: A case study on the Gentoo community. In: International Conference on Cloud and Green Computing, pp. 410–417 (2013)
20. Giraud-Carrier, C., Dunham, M.H.: On the importance of sharing negative results. SIGKDD Explor. Newsl. **12**(2), 3–4 (2011). DOI 10.1145/1964897.1964899. URL `http://doi.acm.org.library.sutd.edu.sg:2048/10.1145/1964897.1964899`
21. Goul, M., Marjanovic, O., Baxley, S., Vizecky, K.: Managing the Enterprise Business Intelligence App Store: Sentiment Analysis Supported Requirements Engineering. In: 2012 45th Hawaii International Conference on System Science (HICSS), pp. 4168–4177 (2012). DOI 10.1109/HICSS.2012.421
22. Gousios, G.: The GHTorrent dataset and tool suite. In: Proceedings of the 10th Working Conference on Mining Software Repositories, MSR'13, pp. 233–236 (2013). URL `http://dl.acm.org/citation.cfm?id=2487085.2487132`
23. Greiler, M., Herzig, K., Czerwonka, J.: Code ownership and software quality: A replication study. In: Proceedings of the 12th Working Conference on Mining Software Repositories, MSR '15, pp. 2–12. IEEE Press, Piscataway, NJ, USA (2015). URL `http://dl.acm.org.library.sutd.edu.sg:2048/citation.cfm?id=2820518.2820522`
24. Guzman, E., Azócar, D., Li, Y.: Sentiment analysis of commit comments in GitHub: An empirical study. In: MSR, pp. 352–355. ACM, New York, NY, USA (2014)
25. Guzman, E., Bruegge, B.: Towards emotional awareness in software development teams. In: Joint Meeting on Foundations of Software Engineering, pp. 671–674. ACM, New York, NY, USA (2013)
26. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The Weka data mining software: An update. SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)
27. Honkela, T., Izzatdust, Z., Lagus, K.: Text mining for wellbeing: Selecting stories using semantic and pragmatic features. In: Artificial Neural Networks and Machine Learning, Part II, *LNCS*, vol. 7553, pp. 467–474. Springer (2012)
28. Howard, M.J., Gupta, S., Pollock, L.L., Vijay-Shanker, K.: Automatically mining software-based, semantically-similar words from comment-code mappings. In: T. Zimmermann, M.D. Penta, S. Kim (eds.) MSR, pp. 377–386. IEEE Computer Society (2013)
29. Jongeling, R., Datta, S., Serebrenik, A.: Choosing your weapons: On sentiment analysis tools for software engineering research. In: ICSME, pp. 531–535. IEEE (2015)
30. Konietschke, F., Hothorn, L.A., Brunner, E.: Rank-based multiple test procedures and simultaneous confidence intervals. Electronic Journal of Statistics **6**, 738–759 (2012)
31. Kucuktunc, O., Cambazoglu, B.B., Weber, I., Ferhatosmanoglu, H.: A Large-scale Sentiment Analysis for Yahoo! Answers. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM '12, pp. 633–642. ACM, New York, NY, USA (2012). DOI 10.1145/2124295.2124371. URL `http://doi.acm.org/10.1145/2124295.2124371`
32. Leopairote, W., Surarerks, A., Prompoon, N.: Evaluating software quality in use using user reviews mining. In: 2013 10th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp. 257–262 (2013). DOI 10.1109/JCSSE.2013.6567355

33. Lindsey, M.R.: What went wrong?: Negative results from VoIP service providers. In: Proceedings of the 5th International Conference on Principles, Systems and Applications of IP Telecommunications, IPTcomm '11, pp. 13:1–13:3. ACM, New York, NY, USA (2011). DOI 10.1145/2124436.2124453. URL http://doi.acm.org.library.sutd.edu.sg:2048/10.1145/2124436.2124453
34. de Magalhães, C.V.C., da Silva, F.Q.B., Santos, R.E.S.: Investigations about replication of empirical studies in software engineering: Preliminary findings from a mapping study. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14, pp. 37:1–37:10. ACM, New York, NY, USA (2014). DOI 10.1145/2601248.2601289. URL http://doi.acm.org.library.sutd.edu.sg:2048/10.1145/2601248.2601289
35. Mende, T.: Replication of defect prediction studies: Problems, pitfalls and recommendations. In: Proceedings of the 6th International Conference on Predictive Models in Software Engineering, PROMISE '10, pp. 5:1–5:10. ACM, New York, NY, USA (2010). DOI 10.1145/1868328.1868336. URL http://doi.acm.org.library.sutd.edu.sg:2048/10.1145/1868328.1868336
36. Mishne, G., Glance, N.S.: Predicting movie sales from blogger sentiment. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, pp. 155–158 (2006)
37. Mohammad, S.M., Kiritchenko, S., Zhu, X.: NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. arXiv:1308.6242 [cs] (2013). URL http://arxiv.org/abs/1308.6242. ArXiv: 1308.6242
38. Murgia, A., Tourani, P., Adams, B., Ortu, M.: Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In: MSR, pp. 262–271. ACM, New York, NY, USA (2014)
39. Novielli, N., Calefato, F., Lanubile, F.: The challenges of sentiment detection in the social programmer ecosystem. In: Proceedings of the 7th International Workshop on Social Software Engineering, SSE 2015, pp. 33–40. ACM, New York, NY, USA (2015). DOI 10.1145/2804381.2804387. URL http://doi.acm.org/10.1145/2804381.2804387
40. Ortu, M., Adams, B., Destefanis, G., Tourani, P., Marchesi, M., Tonelli, R.: Are bullies more productive? empirical study of affectiveness vs. issue fixing time. In: MSR (2015)
41. Pak, A., Paroubek, P.: Twitter Based System: Using Twitter for Disambiguating Sentiment Ambiguous Adjectives. In: Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval '10, pp. 436–439. Association for Computational Linguistics, Stroudsburg, PA, USA (2010). URL http://dl.acm.org/citation.cfm?id=1859664.1859761
42. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval **2**(1-2), 1–135 (2007)
43. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP '02, pp. 79–86. Association for Computational Linguistics, Stroudsburg, PA, USA (2002). DOI 10.3115/1118693.1118704. URL http://dx.doi.org/10.3115/1118693.1118704
44. Panichella, S., Sorbo, A.D., Guzman, E., Visaggio, C.A., Canfora, G., Gall, H.C.: How can I improve my app? classifying user reviews for software maintenance and evolution. In: ICSME, pp. 281–290. IEEE (2015)
45. Pletea, D., Vasilescu, B., Serebrenik, A.: Security and emotion: Sentiment analysis of security discussions on GitHub. In: MSR, pp. 348–351. ACM, New York, NY, USA (2014)
46. Pritchard, P.: Some negative results concerning prime number generators. Commun. ACM **27**(1), 53–57 (1984). DOI 10.1145/69605.357970. URL http://doi.acm.org.library.sutd.edu.sg:2048/10.1145/69605.357970
47. Rousinopoulos, A.I., Robles, G., González-Barahona, J.M.: Sentiment analysis of Free/Open Source developers: preliminary findings from a case study. Revista Eletrônica de Sistemas de Informação **13**(2), 6:1–6:21 (2014)
48. Santos, J.M., Embrechts, M.: On the use of the adjusted rand index as a metric for evaluating supervised classification. In: International Conference on Artificial Neural Networks, *LNCS*, vol. 5769, pp. 175–184. Springer (2009)
49. Schröter, A., Aranda, J., Damian, D., Kwan, I.: To talk or not to talk: factors that influence communication around changesets. In: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, CSCW '12, p. 13171326. ACM, New York, NY, USA (2012). DOI 10.1145/2145204.2145401. URL http://doi.acm.org/10.1145/2145204.2145401

50. Sfetsos, P., Adamidis, P., Angelis, L., Stamelos, I., Deligiannis, I.: Investigating the Impact of Personality and Temperament Traits on Pair Programming: A Controlled Experiment Replication. In: Quality of Information and Communications Technology (QUATIC), 2012 Eighth International Conference on the, pp. 57–65 (2012). DOI 10.1109/QUATIC.2012.36
51. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures, 4 edn. Chapman & Hall (2007)
52. Shihab, E., Kamei, Y., Bhattacharya, P.: Mining challenge 2012: The Android platform. In: MSR, pp. 112–115 (2012)
53. Shull, F.J., Carver, J.C., Vegas, S., Juristo, N.: The role of replications in empirical software engineering. Empirical Software Engineering **13**(2), 211–218 (2008). DOI 10.1007/s10664-008-9060-1. URL http://dx.doi.org/10.1007/s10664-008-9060-1
54. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Empirical Methods in Natural Language Processing, pp. 1631–1642. Ass. for Comp. Linguistics (2013)
55. Täht, D.: The value of repeatable experiments and negative results: - a journey through the history and future of aqm and fair queuing algorithms. In: Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop, CSWS '14, pp. 1–2. ACM, New York, NY, USA (2014). DOI 10.1145/2630088.2652480. URL http://doi.acm.org.library.sutd.edu.sg:2048/10.1145/2630088.2652480
56. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. J. Am. Soc. Inf. Sci. Technol. **63**(1), 163–173 (2012)
57. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment in short strength detection informal text. J. Am. Soc. Inf. Sci. Technol. **61**(12), 2544–2558 (2010)
58. Tonella, P., Torchiano, M., Du Bois, B., Systä, T.: Empirical studies in reverse engineering: State of the art and future trends. Empirical Softw. Engg. **12**(5), 551–571 (2007). DOI 10.1007/s10664-007-9037-5. URL http://dx.doi.org.library.sutd.edu.sg:2048/10.1007/s10664-007-9037-5
59. Tourani, P., Jiang, Y., Adams, B.: Monitoring Sentiment in Open Source Mailing Lists: Exploratory Study on the Apache Ecosystem. In: Proceedings of 24th Annual International Conference on Computer Science and Software Engineering, CASCON '14, pp. 34–44. IBM Corp., Riverton, NJ, USA (2014). URL http://dl.acm.org/citation.cfm?id=2735522.2735528
60. Tukey, J.W.: Quick and dirty methods in statistics, part II, Simple analysis for standard designs. In: American Society for Quality Control, pp. 189–197 (1951)
61. Tumasjan, A., Sprenger, T.O., Sandner, P.G., Welpe, I.M.: Predicting elections with twitter: What 140 characters reveal about political sentiment. In: International AAAI Conference on Weblogs and Social Media, pp. 178–185 (2010)
62. Vasilescu, B., Serebrenik, A., Goeminne, M., Mens, T.: On the variation and specialisation of workload – a case study of the Gnome ecosystem community. Empirical Software Engineering **19**(4), 955–1008 (2013). DOI http://dx.doi.org/10.1007/s10664-013-9244-1
63. Vivian, R., Tarmazdi, H., Falkner, K., Falkner, N., Szabo, C.: The Development of a Dashboard Tool for Visualising Online Teamwork Discussions. In: Proceedings of the 37th International Conference on Software Engineering - Volume 2, ICSE '15, pp. 380–388. IEEE Press, Piscataway, NJ, USA (2015). URL http://dl.acm.org/citation.cfm?id=2819009.2819070
64. Wang, S., Lo, D., Vasilescu, B., Serebrenik, A.: EnTagRec: An enhanced tag recommendation system for software information sites. In: ICSME, pp. 291–300. IEEE (2014)
65. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin **1**(6), 80–83 (1945)
66. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Human Language Technology and Empirical Methods in Natural Language Processing, pp. 347–354. Association for Computational Linguistics, Stroudsburg, PA, USA (2005)
67. Zimmerman, D.W., Zumbo, B.D.: Parametric alternatives to the Student t test under violation of normality and homogeneity of variance. Perceptual and motor skills **74**(3(1)), 835–844 (1992)