Empir Software Eng (2017) 22:2543–2584
DOI 10.1007/s10664-016-9493-x

On negative results when using sentiment analysis tools for software engineering research

Robbert Jongeling[1] · Proshanta Sarkar[2] · Subhajit Datta[3] · Alexander Serebrenik[1]

Published online: 10 January 2017

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 diverging 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.

Communicated by: Richard Paige, Jordi Cabot and Neil Ernst

✉ Alexander Serebrenik
a.serebrenik@tue.nl

Robbert Jongeling
r.m.jongeling@alumnus.tue.nl

Proshanta Sarkar
proshant.cse@gmail.com

Subhajit Datta
subhajit.datta@acm.org

[1] Eindhoven University of Technology, Eindhoven, The Netherlands

[2] IBM India Private Limited, Kolkata, India

[3] Singapore University of Technology and Design, Singapore, Singapore

Keywords Sentiment analysis tools · Replication study · Negative results

1 Introduction

Sentiment analysis is "the task of identifying positive and negative opinions, emotions, and evaluations" (Wilson et al. 2005). 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 (Honkela et al. 2012), predict the outcome of an election (Tumasjan et al. 2010) or movie sales (Mishne and Glance 2006). The spectrum of sentiment analysis techniques ranges from identifying polarity (positive or negative) to a complex computational treatment of subjectivity, opinion and sentiment (Pang and Lee 2007). In particular, the research on sentiment polarity analysis has resulted in a number of mature and publicly available tools such as SentiStrength (Thelwall et al. 2010), Alchemy,[1] Stanford NLP sentiment analyser (Socher et al. 2013) and NLTK (Bird et al. 2009).

In recent times, large scale software development has become increasingly social. With the proliferation of collaborative development environments, discussion between developers are recorded and archived to an extent that could not be conceived before. The availability of such discussion materials makes it easy to study whether and how the sentiments expressed by software developers influence the outcome of development activities. With this background, we apply sentiment polarity analysis to several software development ecosystems in this study.

Sentiment polarity analysis has been recently applied in the software engineering context to study commit comments in GitHub (Guzman et al. 2014), GitHub discussions related to security (Pletea et al. 2014), productivity in Jira issue resolution (Ortu et al. 2015), activity of contributors in Gentoo (Garcia et al. 2013), classification of user reviews for maintenance and evolution (Panichella et al. 2015) and evolution of developers' sentiments in the openSUSE Factory (Rousinopoulos et al. 2014). It has also been suggested when assessing technical candidates on the social web (Capiluppi et al. 2013). Not surprisingly, all the aforementioned software engineering studies with the notable exception of the work by Panichella et al. (2015), reuse the existing sentiment polarity tools, e.g., (Pletea et al. 2014) and (Rousinopoulos et al. 2014) use NLTK, while (Garcia et al. 2013; Guzman and Bruegge 2013; Guzman et al. 2014; Novielli et al. 2015) and (Ortu et al. 2015) 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 (Guzman et al. 2014; Pletea et al. 2014).

Therefore, in this paper we focus on sentiment polarity analysis (Wilson et al. 2005) and investigate to what extent are the software engineering results obtained from sentiment analysis depend on the choice of the sentiment analysis tool. We recognize that there are multiple ways to measure outcomes in software engineering. Among them, time to resolve a particular defect, and/or respond to a particular query are relevant for end users. Accordingly, in

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

the different data-sets studied in this paper, we have taken such resolution or response times to reflect the outcomes of our interest.

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 results from 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 analysis 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. The next section outlines the sentiment analysis tools we have considered in this study. In Section 3 we study agreement between the tools and the results of manual labeling, and between the tools themselves, i.e., *RQ1* and *RQ2*. In Section 4 we conduct a series of studies 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 5 we perform replication of two published studies answering *RQ4* and establishing that conclusions of previously published works cannot be reproduced when a different sentiment analysis tool is used. Finally, in Section 6 we discuss related work and conclude in Section 7.

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

2 Sentiment Analysis Tools

2.1 Tool Selection

To perform the tool evaluation we have decided to focus on open-source tools. This requirement excludes such commercial tools as Lymbix[3] Sentiment API of MeaningCloud[4] or

[2] http://ow.ly/HvC5302N4oK

[3] http://www.lymbix.com/supportcenter/docs

[4] https://www.meaningcloud.com/developer/sentiment-analysis

GetSentiment.[5] Furthermore, we exclude tools that require training before they can be applied such as LibShortText (Yu et al. 2013) or sentiment analysis libraries of popular machine learning tools such as RapidMiner or Weka. Finally, since the software engineering texts that have been analyzed in the past can be quite short (JIRA issues, STACK OVERFLOW questions), we have chosen tools that have already been applied either to software engineering texts (SENTISTRENGTH and NLTK) or to short texts such as tweets (Alchemy or Stanford NLP sentiment analyser).

2.2 Description of Tools

2.2.1 SENTISTRENGTH

SENTISTRENGTH is the sentiment analysis tool most frequently used in software engineering studies (Garcia et al. 2013; Guzman et al. 2014; Novielli et al. 2015; Ortu et al. 2015). Moreover, SENTISTRENGTH had the highest average accuracy among fifteen Twitter sentiment analysis tools (Abbasi et al. 2014). 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.

Interpretation In order to map the separate positivity and negativity scores to a sentiment (positive, neutral or negative) for an entire text fragment, we follow the approach by Thelwall et al. (2012). 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.

2.2.2 Alchemy

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).[6] The sentiment analysis 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. The *status* reflects the analysis success and it is either "OK" or "ERROR".

Interpretation We ignore texts with status "ERROR" or a non-English language. For the remaining texts we consider them as being negative, neutral or positive as indicated by the returned type.

2.2.3 NLTK

NLTK has been applied in earlier software engineering studies (Pletea et al. 2014; Rousinopoulos et al. 2014). NLTK uses a simple bag of words model and returns for each

[5] https://getsentiment.3scale.net/

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

text three probabilities: a probability of the text being negative, one of it being neutral and one of it being positive. To call NLTK, we use the API provided at text-processing.com.[7]

Interpretation If the probability score for neutral is greater than 0.5, the text is considered neutral. Otherwise, it is considered to be the other sentiment with the highest probability (Pletea et al. 2014).

2.2.4 Stanford NLP

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, e.g., in NLTK. Indeed, Socher et al. argue that such an analysis should outperform the bag of words model on short texts (Socher et al. 2013). The 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.

Interpretation To determine 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.

3 Agreement Between Sentiment Analysis Tools

In this section we address *RQ1* and *RQ2*, i.e., to what extent do the different sentiment analysis tools described earlier, 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 (Murgia et al. 2014).

3.1 Methodology

3.1.1 Manually-Labeled Software Engineering Data

As the "golden set" we use the data from a developer emotions study by Murgia et al. (2014). In this study, four evaluators manually labeled 392 comments with emotions "joy", "love", "surprise", "anger", "sadness" or "fear". Emotions "joy" and"love" are taken as indicators of positive sentiments and "anger", "sadness" and "fear"—of negative sentiment. We exclude information about the "surprise" sentiment, since surprises can be, in general, both positive and negative depending on the expectations of the speaker.

We focus on consistently labeled comments. We consider the comment as positive if at least three evaluators have indicated a positive sentiment and no evaluator has indicated negative sentiments. Similarly, we consider the comment as negative if at least three evaluators have indicated a negative sentiment and no evaluator has indicated positive sentiments. Finally, a text is considered as neutral when three or more evaluators have neither indicated a positive sentiment nor a negative sentiment.

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

Using these rules we can conclude that 265 comments have been labeled consistently: 19 negative, 41 positive and 205 neutral. The remaining $392 - 265 = 127$ comments from the study Murgia et al. (2014) have been labeled with contradictory labels e.g. "fear" by one evaluator and "joy" by another.

3.1.2 Evaluation Metrics

Since more than 77 % of the comments have been manually labeled as neutral, i.e., the dataset is unbalanced, traditional metrics such as accuracy might be misleading (Batista et al. 2000): 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, rather than reporting accuracy of the approaches we use the Weighted kappa (Cohen 1968) and the Adjusted Rand Index (ARI) (Hubert and Arabie 1985; Santos and Embrechts 2009). For the sake of completeness we report the F-measures for the three categories of sentiments.

Kappa is a measure of interrater agreement. As recommended by Bakeman and Gottman (Bakeman and Gottman 1997, p. 66) we opt for the *weighted* kappa (κ) since the sentiments can be seen as ordered, from positive through neutral to negative, and disagreement between positive and negative is more "severe" than between positive and neutral or negative and neutral. Our weighting scheme, also following the guidelines of Bakeman and Gottman, is shown in Table 1. We follow the interpretation of κ as advocated by Viera and Garrett (Viera and Garrett 2005) since it is more fine grained than, e.g., the one suggested by Fleiss et al. (2003, p. 609). We say that the agreement is less than chance if $\kappa \leq 0$, slight if $0.01 \leq \kappa \leq 0.20$, fair if $0.21 \leq \kappa \leq 0.40$, moderate if $0.41 \leq \kappa \leq 0.60$, substantial if $0.61 \leq \kappa \leq 0.80$ and almost perfect if $0.81 \leq \kappa \leq 1$. To answer the first research question we look for the agreement between the tool and the manual labeling; to answer the second one—for agreement between two tools.

ARI measures the correspondence between two partitions of the same data. Similarly to the Rand index (Rand 1971), ARI evaluates whether pairs of observations (comments) are considered as belonging to the same category (sentiment) rather than on whether observations (comments) have been assigned to correct classes (sentiment). As opposed to the Rand index, ARI corrects for the possibility that pairs of observations have been put in the same category by chance. 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. To answer the first research question we look for the 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.

Finally, F-measure, introduced by Lewis and Gale (1994) based on the earlier E-measure of Van Rijsbergen (1979, p. 128), is the harmonic mean of the precision and recall. Recall that precision in the classification context is the ratio of true positives[8] and all entities predicted to be positive, while recall is the ratio of true positives and all entities known to be positive. The symmetry between precision and recall, false positives and false negatives, inherent in the F-measure makes it applicable both when addressing *RQ1* and when addressing *RQ2*. We report the F-measure separately for the three classes: neutral, positive and negative.

[8]Here "positive" is not related to the positive sentiment.

Table 1 Weighting scheme for the weighted kappa computation

	positive	neutral	negative
positive	0	1	2
neutral	1	0	1
negative	2	1	0

3.2 Results

None of the 265 consistently labeled comments produce SentiStrength results with $p = -n$ and $p \geq 4$. Three comments produce the "ERROR" status with Alchemy; those comments have been excluded from consideration. We exclude those comments from consideration and report κ and ARI for 262 comments.

Results obtained both for *RQ1* and for *RQ2* are summarized in Table 2 . Detailed confusion matrices relating the results of the tools and the manual labeling as well as results of different tools to each other are presented in Appendix A.

3.3 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.

RQ1 As can be observed from Table 2 both κ and ARI show that the tools are quite far from agreeing with the manual labeling: κ is merely fair, and ARI is low. NLTK scores best, followed by SentiStrength, and both perform better than Alchemy and Stanford NLP. Even when focusing solely on the positive and the negative sentiment, the F-values suggest that improving the F-value for the negative sentiments tends to decrease the F-value for the positive ones, and *vice versa*.

RQ2 Values of κ and ARI obtained when different tools have been compared are even lower when compared to the results of the agreement with the manual labeling. The highest

Table 2 Agreement of sentiment analysis tools with the manual labeling and with each other

Tools			κ	ARI	F neu	F pos	F neg
NLTK	vs.	manual	0.33	0.21	0.76	0.53	0.31
SentiStrength	vs.	manual	0.31	0.13	0.73	0.47	0.35
Alchemy	vs.	manual	0.26	0.07	0.53	0.54	0.23
Stanford NLP	vs.	manual	0.20	0.11	0.48	0.53	0.20
NLTK	vs.	SentiStrength	0.22	0.08	0.64	0.45	0.33
NLTK	vs.	Alchemy	0.20	0.09	0.52	0.60	0.44
NLTK	vs.	Stanford NLP	0.12	0.05	0.48	0.42	0.47
SentiStrength	vs.	Alchemy	0.07	0.07	0.56	0.55	0.38
SentiStrength	vs.	Stanford NLP	−0.14	0.00	0.51	0.33	0.35
Alchemy	vs.	Stanford NLP	0.25	0.05	0.41	0.43	0.58

value of κ, 0.25, has been obtained for Alchemy and Stanford NLP, and is only fair. Agreement between NLTK and SENTISTRENGTH is, while also only fair, the second highest one among the six possible pairs in Table 2.

To illustrate the reasons for the disagreement between the tools and the manual labeling as well as between the tools themselves we discuss a number of example comments.

Example 1 Our first example is a developer describing a clearly undesirable behavior (memory leak) in Apache UIMA. The leak, however, has been fixed; the developer confirms this and thanks the community.

> To test this I used an aggregate AE with a CAS multiplier that declared getCasInstancesRequired()=5. If this AE is instantiated and run in a loop with earlier code it eats up roughly 10MB per iteration. No such leak with the latest code. Thanks!

Due to presence of the expression of gratitude, the comment has been labeled as "love" by all four participants of the Murgia's study. We interpret this as a clear indication of the positive sentiment. However, none of the tools is capable of recognizing this: SENTISTRENGTH labels the comment as being neutral, NLTK, Alchemy and Stanford NLP—as being negative. Indeed, for instance Stanford NLP believes the first three sentences to be negative (e.g., due to presence of "No"), and while it correctly recognizes the last sentence as positive, this is not enough to change the evaluation of the comment as the whole.

Example 2 The following comment from Apache Xerces merely describes an action that has taken place ("committed a patch").

> D.E. Veloper[9] committed your patch for Xerces 2.6.0. Please verify.

Three out of four annotators do not recognize presence of emotion in this comment and we interpret this as the comment being neutral. However, keyword-based sentiment analysis tools might wrongly identify presence of sentiment. For instance, in SentiWordNet (Baccianella et al. 2010) the verb "commit", in addition to neutral meanings (e.g., perpetrate an act as in "commit a crime") has several positive meanings (e.g., confer a trust upon, "I commit my soul to God" or cause to be admitted when speaking of a person to an institution, "he was committed to prison"). In a similar way, the word "patch", in addition to neutral meanings, has negative meanings (e.g.,, *sewing that repairs a worn or torn hole* or *a piece of soft material that covers and protects an injured part of body*). Hence, it should come as no surprise that some sentiment analysis tools identify this comment as positive, some other as negative and finally, some as neutral.

These examples show that in order to be successfully applied in the software engineering context, sentiment analysis tools should become aware of the peculiarities of the software engineering domain: e.g., that words "commit" and "patch" are merely technical terms and do not express sentiment. Our observation concurs with the challenge Novielli et al. (2015) has recognized in sentiment detection in the social programming ecosystem such as STACK OVERFLOW.

[9]To protect the privacy of the project participants we do not disclose their names.

Table 3 Agreement of groups of tools with the manual labeling (n—the number of comments the tools agree upon)

Tools	n	κ	ARI	F		
				neu	pos	neg
NLTK, SentiStrength	138	0.65	0.51	0.89	0.78	0.56
NLTK, Alchemy	134	0.46	0.24	0.73	0.69	0.47
NLTK, Stanford NLP	122	0.43	0.23	0.71	0.74	0.40
SentiStrength, Alchemy	133	0.50	0.27	0.76	0.71	0.43
SentiStrength, Stanford NLP	109	0.53	0.34	0.78	0.83	0.39
Alchemy, Stanford NLP	130	0.36	0.19	0.49	0.79	0.31
NLTK, SentiStrength, Alchemy	88	0.68	0.49	0.84	0.84	0.58
NLTK, SentiStrength, Stanford NLP	71	0.72	0.52	0.85	0.91	0.55
SentiStrength, Alchemy, Stanford NLP	74	0.59	0.38	0.73	0.91	0.41
NLTK, Alchemy, Stanford NLP	75	0.55	0.28	0.68	0.83	0.52
NLTK, SentiStrength, Alchemy, Stanford NLP	53	0.72	0.50	0.80	0.93	0.57

3.4 A Follow-up Study

Given the disagreement between different sentiment analysis tools, we wonder whether focusing only on the comments where the tools agree with each other, would result in a better agreement with the manual labeling. Clearly, since the tools tend to disagree, such a focus reduces the number of comments that can be evaluated. However, it is *a priori* not clear whether a better agreement can be expected with the manual labeling. Thus, we have conducted a follow-up study: for every group of tools we consider only comments on which the tools agree, and determine κ, ARI and the F-measures with respect to the manual labeling.

Results of the follow up study are summarized in Table 3. As expected, the more tools we consider the less comments remain. Recalling that in our previous evaluation 262 comments have been considered, only 52.6 % remain if agreement between two tools is required. For four tools slightly more than 20 % of the comments remain. We also see that focusing on the comments where the tools agree improves the agreement with the manual labeling both in terms of κ and in terms of ARI. The F-measures follow, in general, the same trend. This means a trade-off should be sought between the number of comments the tools agree upon and the agreement with the manual labeling.

3.5 Threats to Validity

As any empirical evaluation, the study presented in this section is subject to threats to validity:

- *Construct validity* might have been threatened by our operationalization of sentiment polarity via emotion, recorded in the dataset by Murgia et al. (2014) (cf. the observations of Novielli et al. (2015)).
- *Internal validity* of our evaluation might have been affected by the exact ways tools have been applied and the interpretation of the tools' output as indication of sentiment,

e.g., calculation of a document-level sentiment as $-2 * \#0 - \#1 + \#3 + 2 * \#4$ for Stanford NLP. Another threat to internal validity stems form the choice of the evaluation metrics: to reduce this threat we report several agreement metrics (ARI, weighted κ and F-measures) recommended in the literature.

– *External validity* of this study can be threatened by the fact that only one dataset has been considered and by the way this dataset has been constructed and evaluated by Murgia et al. (2014). To encourage replication of our study and evaluation of its external validity we make publicly available both the source code and the data used to obtain the results of this paper.[10]

3.6 Summary

We have observed that the sentiment analysis tools do not agree with the manual labeling (*RQ1*) and neither do they agree with each other (*RQ2*).

4 Impact of the Choice of Sentiment Analysis Tool

In Section 3 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, we now address *RQ3* and discuss a simple set-up of a study aiming at understanding differences in response times for positive, neutral and negative texts.

4.1 Methodology

We study whether differences can be observed between response times (issue resolution times or question answering times) for positive, neutral and negative texts in the context of addressing *RQ3*. 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.

Recent studies considering sentiment in software engineering data tend to include additional variables, e.g., sentiment analysis has been recently combined with politeness analysis (Danescu-Niculescu-Mizil et al. 2013) to study issue resolution time (Destefanis et al. 2016; Ortu et al. 2015). To illustrate the impact of the choice of sentiment analysis tool on the study outcome in presence of other analysis techniques, we repeat the response time study but combine sentiment analysis with politeness analysis.

4.1.1 Sentiment Analysis Tools

Based on the answers to *RQ1* and *RQ2* presented in Section 3.3 we select SentiStrength and NLTK to address *RQ3*. Indeed, NLTK scores best when compared to the manual

[10] http://ow.ly/HvC5302N4oK

Table 4 Descriptive statistics of resolution/response type

	Mean	Std Dev	Median
Android	79.58	143.19	9
Gnome	267.03	1.33	26.94
SO	21.53	131.32	0.13
ASF	96.57	255.44	4.16

labelling, followed by SentiStrength, and both perform better than Alchemy and Stanford NLP. Agreement between NLTK and SentiStrength, while also only fair, is still the second highest one among the six possible pairs in Table 2.

Moreover, we also repeat each study on the subset of texts where NLTK and SentiStrength agree. Indeed, Table 3 shows that these tools agree upon the largest subset of comments, achieving at the same time the highest among the two-tool combinations κ, ARI and the F-measure for neutral and negative class. We also observe that further improvement of the evaluation metrics is possible but at cost of significant drop in the number of comments.

4.1.2 Datasets

We study 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. To ensure validity of our study we have opted for five datasets collected independently by other researchers (Android Issue Tracker descriptions and titles, Gnome Issue Tracker descriptions, ASF Issue Tracker descriptions and titles) and two dataset derived by us from a well-known public data source (Gnome-Related Stack Overflow question titles and bodies). All datasets are publicly available for replication purposes.[11] The descriptive statistics of the resolution/response times from these data-sets are given in Table 4.

Android Issue Tracker A dataset of 20,169 issues from the Android issue tracker was part of the mining challenge of MSR 2012 (Shihab et al. 2012). 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.[12] Similarly to the Android dataset, we have looked only at issues with a value for field *bug_status* of `resolved`. In total

[11] http://ow.ly/HvC5302N4oK

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

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.

GNOME-Related STACK OVERFLOW Discussions We use the StackExchange online data explorer[13] 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. (2015) 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*.

4.1.3 Politeness Analysis

Similarly to sentiment analysis classifying texts into positive, neutral and negative, politeness analysis classifies texts into polite, neutral and impolite. In our work we use the Stanford politeness API[14] based on the work of Danescu-Niculescu-Mizil et al. (2013). As opposed to sentiment analysis tools such as SENTISTRENGTH and NLTK, the Stanford politeness API has been evaluated on software engineering data: STACK OVERFLOW questions and answers.

Given a textual fragment the Stanford politeness API returns a politeness score ranging between 0 (impolite) and 1 (polite) with 0.5 representing the "ideal neutrality". To discretize the score into polite, neutral and impolite we apply the Stanford politeness API to the seven datasets above. It turns out that the politeness scores of the majority of comments are low: the median score is 0.314, the mean score is 0.361 and the third quartile (Q3) is 0.389. We use the latter value to determine the neutrality range. We say therefore that the comments scoring between 0.389 and $0.611 = 1 - 0.389$ are neutral; comments scoring lower than 0.389 are impolite and comments scoring higher than 0.611 are polite.

4.1.4 Statistical Analysis

To answer our research questions we need to compare distributions of response times corresponding to different groups of issues. We conduct two series of studies. In the first series of studies we compare the distributions of the response times corresponding to positive, neutral and negative questions/issues. In the second series we also consider politeness and compare the distributions of the response times corresponding to nine groups obtained through all possible combinations of sentiment (positive, neutral and negative) and politeness (polite, neutral and impolite).

[13] http://data.stackexchange.com/

[14] https://github.com/sudhof/politeness

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 (Wilcoxon 1945), with correction for multiple comparisons, e.g., Bonferroni correction (Dunn 1961; Sheskin 2007). Unfortunately, the global test null hypothesis may be rejected while none of the sub-hypotheses are rejected, or vice versa (Gabriel 1969). 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 (Brunner and Munzel 2000; Zimmerman and Zumbo 1992). 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 (Konietschke et al. 2012) for Tukey-type contrasts (Tukey 1951), the probit transformation and the traditional 5 % family error rate (cf. Vasilescu et al. 2013; Wang et al. 2014).

The results of the $\widetilde{\mathbf{T}}$-procedure are a series of probability estimates $p(a, b)$ with the corresponding p-values, where a and b are representing the distributions being compared. 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. Ortu et al. 2015) since it allows for an easy comparison of the results obtained for different tools.

4.1.5 Agreement Between the Results

Recall that sentiment analysis tools induce partition of the response times into categories. For every pair of values (a, b) the $\widetilde{\mathbf{T}}$-procedure indicates one of the three following outcomes: $>$ (response times in category a tends to be larger than those in category b), $<$ (response times in category b tends to be larger than those in category a) or $\parallel$ (no evidence has been found for difference in response times corresponding to categories a and b). We stress that we refrain from interpreting lack of evidence for difference as evidence for lack of difference, i.e., we do not claim the distributions of response times corresponding to categories a and b are the same but merely that we cannot find evidence that these distributions are not the same. Hence, we also use $\parallel$ (incomparable) rather than $=$ (equal).

To compare the tools we therefore need to assess the agreement between the results produced by the $\widetilde{\mathbf{T}}$-procedure for partitions induced by different tools.

Example 3 Let $\widetilde{\mathbf{T}}$-procedure report "pos $<$ neu", "pos $<$ neg" and "neu $<$ neg" for partitions induced by Tool1, "pos $<$ neu", "pos $<$ neg" and "neu $\parallel$ neg" for partitions induced by Tool2, and "pos $>$ neu", "pos $>$ neg" and "neu $\parallel$ neg" for partitions induced by Tool3. Then, we would like to say that Tool1 agrees more with Tool2 than with Tool3, and Tool2 agrees more with Tool3 than with Tool1.

Unfortunately, traditional agreement measures such as discussed in Section 3.1.2 are no longer applicable since the number of datapoints (pairs of categories) is small: 3 for

sentiment and 36 for the sentiment-politeness combination. Hence, we propose to count the pairs of categories (a, b) such that the $\widetilde{\mathbf{T}}$-procedure produces the same result for partitions induced by both tools (so called observed agreement).

Example 4 For Example 3 we observe that Tool1 and Tool2 agree on two pairs, Tool1 and Tool3 agree on zero pairs, and Tool2 and Tool3 agree on one pair.

We believe, however, that a disagreement between claims "response times in category a tends to be larger than those in category b" and "response times in category b tends to be larger than those in category a" is more severe than between claims "response times in category a tends to be larger than those in category b" and "no evidence has been found for difference in response times corresponding to categories a and b". One possible way to address this concern would be to associate different kinds of disagreement with different weights: this is an approach taken, e.g., by the weighted κ (Cohen 1968). However, the choice of specific weights might appear arbitrary.

Hence, when reporting disagreement between the tools (cf. Tables 6 and 8 below) we report different kinds of disagreement separately, i.e., we report four numbers $x - y - z - w$, where

- x is the number of pairs for which the tools agree about the relation between the response times ($>>$ or $<<$),
- y is the number of pairs for which the tools agree about the lack of such a relation ($\|\|$),
- z is the number of pairs when one of the tools has established the relation and another one did not ($\| >$, $\| <$, $< \|$ or $> \|$),
- w is the number of pairs when the tools have established *different* relations ($<>$ or $><$).

Example 5 Example 3, continued. We report agreement between Tool1 and Tool2 as $2 - 0 - 0 - 1$, between Tool1 and Tool3 as $0 - 0 - 1 - 2$, and between Tool2 and Tool3 as $0 - 1 - 0 - 2$.

4.2 Results

Results of our study are summarized in Table 5. For the sake of readability the relations found are aligned horizontally. For each dataset and each tool 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 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,

Table 5 Comparison of NLTK and SentiStrength. Thresholds for statistical significance: 0.05 (*), 0.01 (**), 0.001 (***). Exact *p*-values are indicated as subscripts; 0 indicates that the p-value is too small to be computed precisely. For the sake of readability we omit pairs for which no evidence has been found for differences in response times

	NLTK	SentiStrength	NLTK ∩ SentiStrength
	neg-neu-pos	neg-neu-pos	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	1,684-2,435-1,182[a]	893-712-299
	$neu > neg^{***}_{2.79\times10^{-8}}$		$neu > neg^{*}_{2.54\times10^{-2}}$
	$neu > pos^{**}_{5.55\times10^{-3}}$	$neu > pos^{**}_{9.72\times10^{-3}}$	$neu > pos^{***}_{7.53\times10^{-5}}$
		$neg > pos^{***}_{6.32\times10^{-4}}$	$neg > pos^{*}_{3.81\times10^{-2}}$
Gnome			
descr	54,032-291,906-20,380	58,585-293,226-14,507	16,829-24,2780-1,785
	$neg > neu^{***}_{0}$	$neg > neu^{***}_{0}$	$neg > neu^{***}_{0}$
	$pos > neu^{***}_{0}$	$pos > neu^{***}_{0}$	$pos > neu^{***}_{0}$
	$pos > neg^{***}_{0}$		
		$neg > pos^{***}_{0}$	
Stack Overflow			
title	84-285-41	53-330-27	16-240-8
	∅	∅	∅
descr	249-71-90	90-183-137	62-35-42
	∅	$neg > pos^{*}_{3.46\times10^{-2}}$	∅
ASF			
title	19,367-67,948-8,348[b]	24,141-62,016-9,510	6,450-44,818-1,106
		$pos > neu^{***}_{0}$	$pos > neu^{**}_{3.71\times10^{-3}}$
	$pos > neg^{***}_{0}$	$pos > neg^{***}_{2.60\times10^{-12}}$	
descr[c]	30,339-42,540-13,129[d]	29,021-41,043-15,971[e]	10,989-20,940-3,814
	$neg > neu^{***}_{0}$	$neg > neu^{***}_{0}$	$neg > neu^{***}_{0}$
	$pos > neu^{***}_{0}$	$pos > neu^{***}_{0}$	$pos > neu^{***}_{0}$
		$pos > neg^{***}_{5.32\times10^{-13}}$	$pos > neg^{***}_{5.12\times10^{-13}}$

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

[b]The tool reported an error for 4 issues.

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

[d]The tool reported an error for 39 issues.

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

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.

Overall, the agreement between NLTK, SentiStrength and NLTK ∩ SentiStrength reported as described in Section 4.1.5 is summarized in Table 6.

Next we perform a similar study by including the politeness information. Table 7 summarizes the findings for Android. Observe that not a single relation could have been

Table 6 Agreement between NLTK, SentiStrength and NLTK ∩ SentiStrength. See Section 4.1.5 for the explanation of the $x - y - z - w$ notation

	NLTK vs. SentiStrength	NLTK vs. NLTK ∩ SentiStrength	SentiStrength vs. NLTK ∩ SentiStrength
	Android		
title	0 – 3 – 0 – 0	0 – 3 – 0 – 0	0 – 3 – 0 – 0
descr	1 – 0 – 2 – 0	2 – 0 – 1 – 0	2 – 0 – 1 – 0
	Gnome		
desc	2 – 0 – 0 – 1	2 – 0 – 1 – 0	2 – 0 – 1 – 0
	Stack Overflow		
title	0 – 3 – 0 – 0	0 – 3 – 0 – 0	0 – 3 – 0 – 0
desc	0 – 2 – 1 – 0	0 – 3 – 0 – 0	0 – 2 – 1 – 0
	ASF		
title	1 – 1 – 1 – 0	0 – 1 – 2 – 0	1 – 1 – 1 – 0
desc	2 – 0 – 1 – 0	2 – 0 – 1 – 0	3 – 0 – 0 – 0

established both by NLTK and by SentiStrength. Results for Gnome, Stack Overflow and ASF are presented in Tables 18, 19 and 20 in the appendix. Agreement is summarized in Table 8: including politeness increases the number of categories to be compared to nine, and therefore, the number of possible category pairs to $\frac{9*8}{2} = 36$. Table 8 suggests that while the tools tend to agree on the relation or lack thereof between most of the category pairs, the differences between the tools account for the differences in the relations observed in up to 30 % (11/36) of the pairs. Still, differences between the tools leading to contradictory results is relatively rare (two cases in Gnome, one in ASF titles and one in ASF descriptions), the differences tend to manifest as a relation being discovered when only one of the tools is used.

4.3 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 (Hall et al. 2009) or RapidMiner[15] on software engineering data.

A similar approach has been recently taken by Panichella et al. (2015) 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 (Gamon et al. 2005) and the need for text-analysis tools specifically targeting texts related to software engineering (Howard et al. 2013) have been recognized in the past.

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

Table 7 Comparison of NLTK and SentiStrength in combination with politeness for the Android datasets. Thresholds for statistical significance: 0.05 (*), 0.01 (**), 0.001 (***). Exact p-values are indicated as subscripts. Results for Gnome, Stack Overflow and ASF are presented in Tables 18, 19 and 20 in the appendix

	NLTK			SentiStrength			NLTK ∩ SentiStrength		
title									
	neg	neu	pos	neg	neu	pos	neg	neu	pos
imp	948	2872	268	1077	2729	279	297	1935	18
neu	245	693	120	315	652	89	86	432	17
pol	37	23	10	22	32	16	13	14	1
		∅			∅			—[a]	
descr									
	neg	neu	pos	neg	neu	pos	neg	neu	pos
imp	262	220	41	218	236	68	118	110	7
neu	562	530	144	470	515	251	211	229	46
pol	1866	907	684	996	1594	863	564	373	246
				neg.neu > pos.pol$^{**}_{1.40\times10^{-3}}$					
				neg.pol > pos.pol$^{*}_{4.55\times10^{-2}}$					
							neu.imp > neg.pol$^{*}_{4.63\times10^{-2}}$		
							neu.imp > pos.pol$^{**}_{7.20\times10^{-3}}$		
	neu.neu > neg.imp$^{*}_{4.23\times10^{-2}}$								
	neu.neu > neg.pol$^{***}_{1.19\times10^{-5}}$						neu.neu > neg.pol$^{*}_{3.89\times10^{-2}}$		
				neu.neu > pos.pol$^{*}_{3.91\times10^{-2}}$			neu.neu > pos.pol$^{**}_{3.14\times10^{-3}}$		
	neu.pol > neg.pol$^{***}_{8.19\times10^{-4}}$								

[a]nparcomp could not run due to insufficient data points

4.4 Threats to Validity

Validity of the conclusions derived might have been threatened by the choice of the data as well by the choice of the statistical machinery.

To reduce the threats related to the data, we have opted for seven different but similar datasets: the Stack Overflow dataset contains information about questions and answers, Android, Gnome and ASF—information about issues. We expect the conclusions above to be valid at least for other issue trackers and software engineering question & answer platforms. For Android, Gnome and ASF we have reused data collected by other researchers (Shihab et al. (2012), Bird[16] and Ortu et al. (2015), respectively). We believe the threats associated with noise in these datasets are limited as they have been extensively used in the previous studies: e.g., Asaduzzaman et al. (Asaduzzaman et al.) and Martie et al. (Martie et al.) used the Android dataset, Linstead and Baldi (2009) used the Gnome dataset, and Ortu et al. (2015) used the ASF dataset. The only dataset we have collected ourselves is the Stack Overflow dataset, and indeed the usual threats related to completeness of the data (questions can be removed) apply. Furthermore, presence of machine-generated text, e.g., error messages, stack traces or source code, might have affected our results.

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

Table 8 Agreement between NLTK, SentiStrength and NLTK ∩ SentiStrength (politeness information included). See Section 4.1.5 for the explanation of the $x - y - z - w$ notation

	NLTK vs. SentiStrength	NLTK vs. NLTK ∩ SentiStrength	SentiStrength vs. NLTK ∩ SentiStrength
	Android		
title	0 – 36 – 0 – 0	—[a]	—[a]
descr	0 – 30 – 6 – 0	1 – 30 – 5 – 0	1 – 30 – 5 – 0
	Gnome		
desc	14 – 13 – 7 – 2	10 – 15 – 11 – 0	10 – 18 – 8 – 0
	Stack Overflow		
title	0 – 28 – 0 – 0[b]	—[c]	—[c]
desc	0 – 33 – 3 – 0	—[c]	—[c]
	ASF		
title	1 – 24 – 10 – 1	0 – 31 – 5 – 0	0 – 27 – 9 – 0
desc	25 – 3 – 7 – 1	23 – 5 – 8 – 0	23 – 4 – 9 – 0

[a]nparcomp could not run on the results of NLTK ∩ SentiStrength due to insufficient data points.

[b]Since the Stack Overflow dataset is relatively small, not all sentiment/politeness combinations are present in the dataset.

[c]Focus on questions where NLTK and SentiStrength agree reduces the number of combinations present making comparing NLTK ∩ SentiStrength and NLTK not possible. Idem for SentiStrength.

Similarly, to reduce the threats related to the choice of the statistical machinery we opt for the $\widetilde{\mathbf{T}}$-approach (Konietschke et al. 2012) that has been successfully applied in the software engineering context (Dajsuren et al. 2013; Li et al. 2014; Sun et al. 2015; Vasilescu et al. 2013; Vasilescu et al. 2013; Wang et al. 2014; Yu et al. 2016).

5 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 we investigate whether this is indeed the case by performing replication studies (Shull et al. 2008) 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 (Shull et al. 2008). In dependent replications the researchers aim at keeping the experiment the same or very similar to the original one, possibly changing the artifact being studied.

5.1 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 (Gousios 2013). The dataset includes information from the top-10 starred repositories in the most popular programming languages and is not representative of GitHub as a whole[17].

[17]http://ghtorrent.org/msr14.html

The first paper we have chosen to replicate is the one by Pletea et al. (2014). 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 fourth author of the current manuscript has also co-authored the work by Pletea et al. (2014).

The second paper we have chosen to replicate is the one by Guzman et al. (2014). 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. This study was the winner of the MSR 2014 challenge.

5.2 Replication Approach

We aim at performing the exact replication of the studies chosen 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 intend to apply SentiStrength in the replication; since Guzman et al. use SentiStrength, we intend to apply NLTK. However, since the exact collections of comments used in the original studies were no longer available, we had to recreate the datasets ourselves. This lead to minor differences with the number of comments we have found as opposed to those reported in the original studies. Hence, we replicate each study *twice*: first applying the same tool as in the original study to a slightly different data, second applying a different sentiment analysis tool to the same data as in the first replication.

We hypothesize that the differences between applying the same tool to slightly different datasets would be small. However, we expect that we might get different, statistically significant, results in these studies when using a different sentiment analysis tool.

5.2.1 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. Comparison of the sentiment recognition using a sentiment analysis tool (NLTK) with 30 manually labeled security-related commit discussions were mixed. Moreover, it has been observed that the NLTK results were mostly bipolar, having both strong negative and strong positive components. Based on this observations the authors suggest that the security-related discussions are more emotional than non-security related ones.

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. We extend the original comparison with the manually labeled discussions by including the results obtained by SentiStrength.

5.2.2 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

Table 9 Identification of security-related comments and discussions results

Type				Comments	Discussions
Commits	Pletea et al. (2014)		Security	2689 (4.43 %)	1809 (9.84 %)
			Total	60658	18380
	Current study	Before elimination	Security	2509 (4.13 %)	1706 (9.28 %)
			Total	60658	18377
		Excluded SentiStrength		9	32
		Excluded NLTK		0	1
		For further analysis	Security	2509 (4.14 %)	1689 (9.21 %)
			Total	60649	18344
	Pletea et al. (2014)		Security	2689 (4.43 %)	1809 (9.84 %)
			Total	60658	18380
	Current study	Before elimination	Security	1801 (3.28 %)	1091 (11.36 %)
			Total	54892	9601
		Excluded SentiStrength		1	16
		Excluded NLTK		5	0
		For further analysis	Security	1800 (3.28 %)	1081 (11.28 %)
			Total	54886	9585

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.

5.3 Replication Results

Here we present the results of replicating both studies.

5.3.1 Pletea et al.

We start the replication by creating Table 9, 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. As explained above, we have found slightly different numbers of comments and discussions in each category. Most notably we find 180 less security-related comments in commits. However, the percentages of security and non-security related comments and discussions are similar.

To ensure validity of the comparison between NLTK and SentiStrength we have applied both tools to comments and discussions. On several occasions the tools reported an error. We have decided to exclude those cases to ensure that further analysis applies to exactly the same comments and discussions. Hence, in Table 9 we also report the numbers of comments and discussions excluded.

Table 10 Commits sentiment analysis statistics. The largest group per study is typeset in boldface

Type			Negative	Neutral	Positive
Discussions	Pletea et al. (2014)	Security	72.52 %	10.88 %	16.58 %
	NLTK	Rest	52.28 %	20.37 %	25.33 %
	Current study	Security	70.16 %	12.79 %	17.05 %
	NLTK	Rest	52.89 %	21.50 %	25.61 %
	Current study	Security	30.66 %	42.92 %	26.40 %
	SentiStrength	Rest	24.13 %	43.92 %	31.94 %
Comments	Pletea et al. (2014)	Security	55.59 %	23.42 %	20.97 %
	NLTK	Rest	46.94 %	26.58 %	26.47 %
	Current study	Security	55.96 %	22.88 %	21.16 %
	NLTK	Rest	46.89 %	26.61 %	26.50 %
	Current study	Security	32.60 %	46.95 %	20.44 %
	SentiStrength	Rest	22.30 %	50.74 %	26.95 %

Next we apply NLTK and SentiStrength to analyze the sentiment of comments and discussions. Tables 10 and 11 present the results Tables 2 and 3 of the original paper, respectively, and extend them by including results of NLTK and SentiStrength on the current study dataset from Table 9. Inspecting Tables 10 and 11 we observe that the values obtained when using NLTK are close to those reported by Pletea et al., while SentiStrength produces very different results. Indeed, NLTK indicates that comments and discussions, submitted via commits or via pull requests, are predominantly negative, while according to SentiStrength neutral is the predominant classification.

Despite those differences, the original conclusion of Pletea et al. still holds: whether we consider comments or discussions, commits or pull requests, percentage of negative texts among security related texts is higher than among non-security related texts.

Finally, in Table 4 Pletea et al. consider thirty security-related commit discussions and compare evaluation of the security relevance and sentiment as determined by the tools with

Table 11 Pull Requests sentiment analysis statistics. The largest group per study is typeset in boldface

Type			Negative	Neutral	Positive
Discussions	Pletea et al. (2014)	Security	81.00 %	5.52 %	13.47 %
	NLTK	Rest	69.58 %	11.98 %	18.42 %
	Current study	Security	77.61 %	7.03 %	15.36 %
	NLTK	Rest	67.43 %	13.82 %	18.76 %
	Current study	Security	30.80 %	45.51 %	23.68 %
	SentiStrength	Rest	24.15 %	51.17 %	24.67 %
Comments	Pletea et al. (2014)	Security	59.83 %	19.09 %	21.06 %
	NLTK	Rest	50.16 %	26.12 %	23.70 %
	Current study	Security	59.67 %	18.83 %	21.50 %
	NLTK	Rest	49.81 %	26.45 %	23.74 %
	Current study	Security	25.66 %	51.22 %	23.11 %
	SentiStrength	Rest	18.14 %	62.87 %	18.97 %

Table 12 Case study results. Strength of the human-labeled sentiments has been labeled by Pletea et al. on a 5-star scale (Pletea et al. 2014)

Sec. relevance	Discussion (Commit ID)	# sec. keywords	Sec. relevance (human)	NLTK neutral (%)	NLTK negative (%)	NLTK positive (%)	NLTK result	SENTI STRENGTH 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(*****)

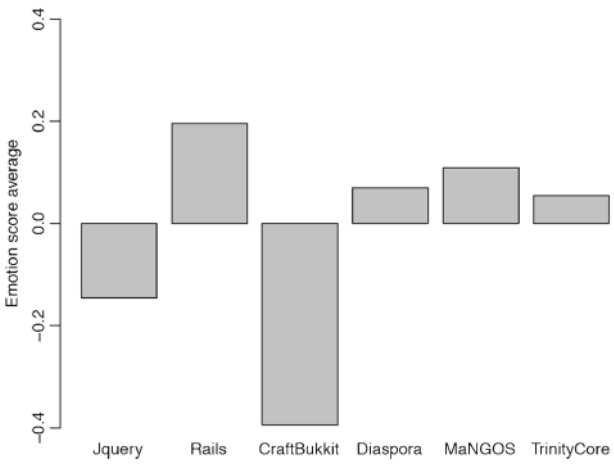

Fig. 1 Emotion score average per project, using SENTISTRENGTH (Guzman et al. 2014)

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 the bottom 10 % of all security-related discussions.

Table 12 extends Table 4 (Pletea et al. 2014) by adding a column with the results of SENTISTRENGTH. Asterisks indicate the strength of the sentiment as perceived by the human evaluator.

By inspecting Table 12 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. We can therefore conclude that replacing NLTK by SENTISTRENGTH did affect the conclusion of the original study: results of the agreement with the manual labeling are still mixed.

We also observe that both for NLTK and for SENTISTRENGTH agreement in the "high" security group is lower than in the "low" security group.

Moreover, Pletea et al. have been observed that the NLTK results were mostly bipolar, having both strong negative and strong positive components, suggesting that security-related discussions are more *emotional*. This observation is not supported by SENTISTRENGTH that classifies 17 out of 30 discussions as neutral.

5.3.2 Guzman et al.

We classified all 60658 commit comments in the MSR 2014 challenge dataset (Gousios 2013) using NLTK.

In the original paper by Guzman et al. (2014) 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.

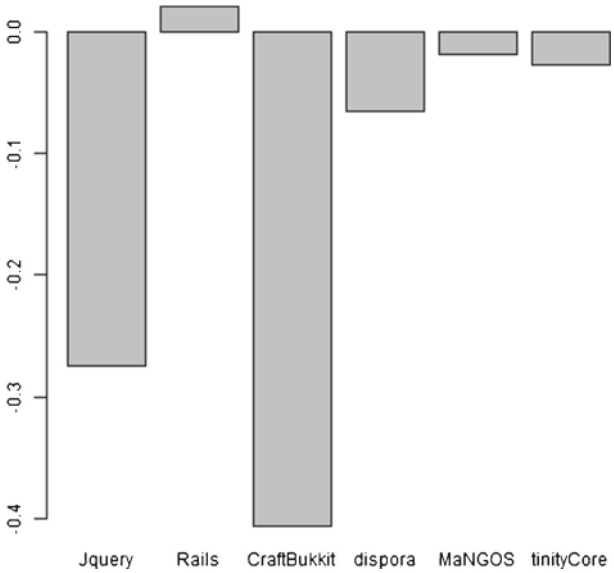

Fig. 2 Proportion of positive, neutral and negative commit comments per project, using SENTISTRENGTH (replication)

Guzman et al. start by considering six projects with the highest number of commit comments: Jquery, Rails, CraftBukkit, Diaspora, MaNGOS and TrinityCore. 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. We replicate their study twice: first of all, using the same tool used by the authors (SENTISTRENGTH), and then using an alternative tool (NLTK).

Figs. 2 and 3 show the replication of the study of the average sentiment score in the six projects. The original figure from the work of Guzman et al. is shown in Fig. 1. Comparing Fig. 1 with Fig. 2 we observe that while the exact values of the averages are lower in the replication, the relative order of the projects is almost the same. Indeed, Rails is the most positive project, followed by MaNGOS and then the close values of Diaspora and TrinityCore, followed by Jquery and at last CraftBukkit. Differences between Figs. 1 and 3 are more pronounced. Indeed, the average emotion score is more negative than in the original study for each project. Moreover, while Jquery and CraftBukkit are still the most negative projects, Rails is no longer positive or even least negative.

Next we consider proportions of negative, neutral and positive sentiments. The original figure from the work of Guzman et al. is shown in Fig. 4, while Figs. 5 and 6 show the results of our replications. NLTK replication (Fig. 6) shows a larger proportion of negative commit comments than in the original paper (Fig. 4), which shows a larger proportion of negative commit comments than the SENTISTRENGTH replication (Fig. 5).

Tables 13–15 contain the results from replicating the studies done in the study by Guzman et al. As above, we replicate those studies twice: using the same tool used by the authors (SENTISTRENGTH), and then using an alternative tool (NLTK).

In contrast to SENTISTRENGTH, NLTK 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,

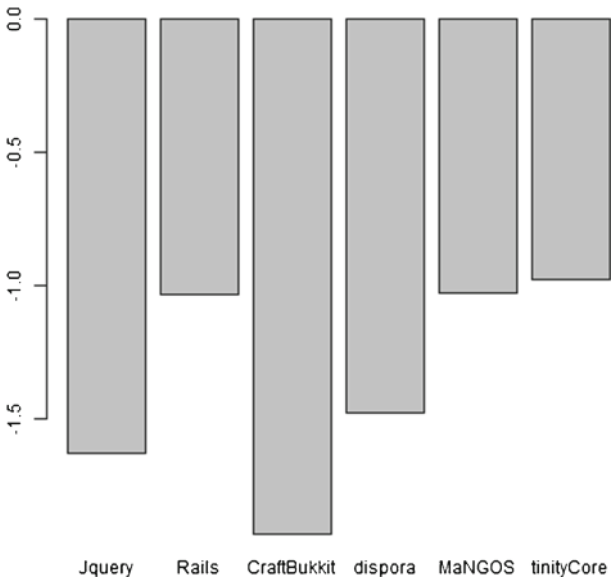

Fig. 3 Emotion score average per project, using NLTK (replication)

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.

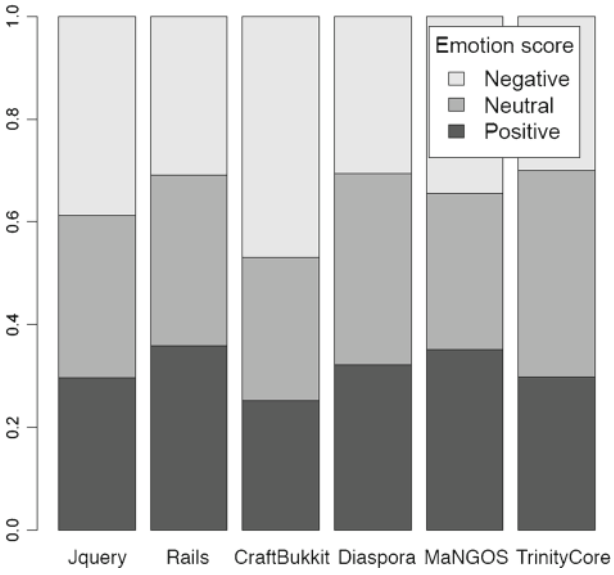

Fig. 4 Proportion of positive, neutral and negative commit comments per project, using SentiStrength (Guzman et al. 2014)

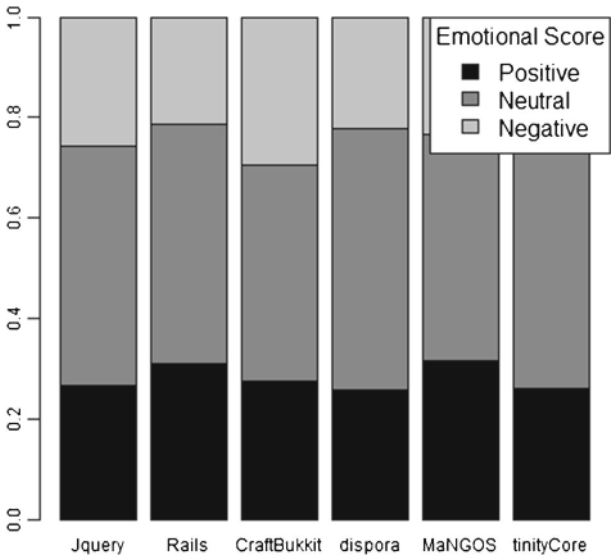

Fig. 5 Proportion of positive, neutral and negative commit comments per project, using SENTISTRENGTH (replication)

Therefore, when using NLTK we apply a transformation to create numbers in the same range according to the following formula:

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

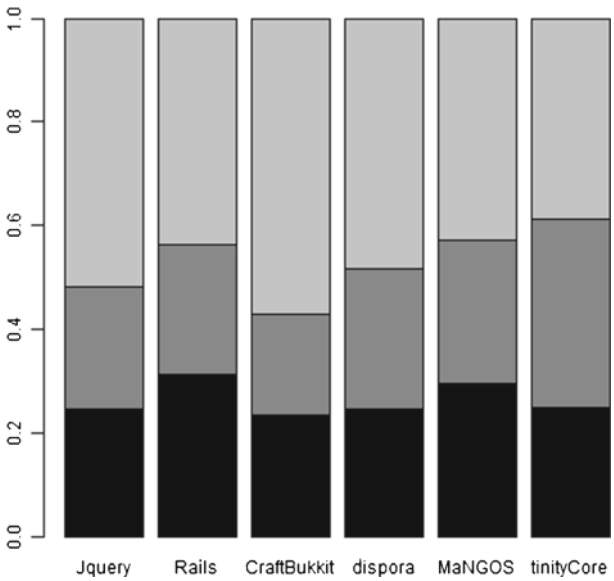

Fig. 6 Proportion of positive, neutral and negative commit comments per project, using NLTK (replication)

Table 13 Emotion score average grouped by programming language

Lang	Guzman et al. (2014)			Current study								
	SentiStrength			Com	SentiStrength				NLTK			
	Com	Mean	SD		Mean	SD	Med	IQR	Mean	SD	Med	IQR
C	6257	0.023	1.716	6277	−0.217	1.746	0.000	2.000	−1.834	3.095	−3.256	4.491
C++	16930	0.017	1.725	16983	−0.031	1.765	0.000	4.000	1.017	2.959	0.000	5.953
Java	4713	−0.144	1.736	4712	−0.282	1.887	0.000	4.000	−1.753	3.106	−3.191	4.460
Python	2128	−0.018	1.711	2133	−0.182	1.709	0.000	2.000	−1.636	3.079	−3.093	4.395
Ruby	15257	0.002	1.714	15355	−0.034	1.794	0.000	4.000	1.243	3.117	0.000	6.293

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 the same data set, and then observe (dis)agreement between the conclusions made. In Tables 13–15 we replicate the sentiment scores grouped by programming language, weekday and time of the day. The original study reports the mean and the standard deviation. However, the mean can be unreliable (Vasilescu et al. 2011) and, therefore, we also report the median and the interquartile range IQR, $Q_3 - Q_1$.

Guzman et al. report that "Java projects tend to have a slightly more negative score than projects implemented in other languages". As can be seen from Table 13, when the same tool (SentiStrength) has been applied to our data set a similar conclusion can be made. This is, however, not the case when NLTK has been applied: Table 13 shows a lower average emotion score for the C programming language than for Java. Also the median score for C is lower than for Java. We can therefore say that validity of this conclusion is not affected by the data set but is affected by the choice of the sentiment analysis tool.

Furthermore, Guzman et al. report that the observation about Java has been statistically confirmed and that the statistical tests on the remaining programming languages (C, C++,

Table 14 Emotion score average grouped by weekday

Day	Guzman et al. (2014)			Current study								
	SentiStrength			Com	SentiStrength				NLTK			
	Com	Mean	SD		Mean	SD	Med	IQR	Mean	SD	Med	IQR
Mon	9517	-0.043	1.732	9533	−0.148	1.790	0.000	4.000	−1.316	3.047	0.000	6.199
Tue	9319	0.005	1.712	9389	−0.089	1.766	0.000	4.000	−1.344	3.079	0.000	6.218
Wed	9730	0.008	1.716	9748	−0.117	1.797	0.000	4.000	−1.372	3.100	0.000	6.292
Thu	9538	0.001	1.728	9561	−0.116	1.791	0.000	4.000	−1.357	3.073	0.000	6.226
Fri	9076	−0.016	1.739	9152	−0.075	1.791	0.000	4.000	−1.347	3.082	0.000	6.256
Sat	6701	−0.027	1.688	6722	− 0.073	1.788	0.000	4.000	−1.326	3.066	0.000	6.264
Sun	6544	0.022	1.717	6544	−0.123	1.774	0.000	4.000	−1.381	3.081	0.000	6.245

Table 15 Emotion score average grouped by time of the day

Day	Guzman et al. (2014)			Current study								
	SentiStrength			Com	SentiStrength				NLTK			
	Com	Mean	SD		Mean	SD	Med	IQR	Mean	SD	Med	IQR
morning	12714	0.001	1.730	12750	−0.112	1.777	0.000	4.000	−1.398	3.062	0.000	6.234
afternoon	19809	0.004	1.717	19859	−0.089	1.764	0.000	4.000	−1.326	3.076	0.000	6.235
evening	16584	−0.023	1.721	16634	−0.102	1.794	0.000	4.000	−1.323	3.085	0.000	6.261
night	11318	−0.016	1.713	11415	−0.142	1.820	0.000	4.000	−1.370	3.077	0.000	6.246

JavaScript, PHP, Python and Ruby) did not yield significant results. The statistical test used is the Wilcoxon rank sum test. The authors compare seven programming languages and report that the corresponding p-values are less or equal to 0.002. We conjecture that the Bonferroni correction for multiple comparisons has been applied since $0.05/21 \simeq 0.0024$.

When replicating this study we first of all exclude projects developed in languages other than the seven languages considered in the original study, and keep 55405 commit comments. Next we compare distributions corresponding to different programming languages. A more statistically sound procedure would have been the $\widetilde{\mathbf{T}}$-procedure discussed in Section 4.1.4. However, in order to keep our replication as close as possible to the original study, we also perform a series of pairwise Wilcoxon tests with the Bonferroni correction.

In the replication with SentiStrength we observe that (1) the claim that Java has more negative score than other languages is not confirmed (p-value for the (Java, C) pair is 0.6552) and (2) lack of statistically significant relation between other programming languages is not confirmed either (e.g., p-value for (C,C++) with the two-sided alternative is 6.9×10^{-12}). Similarly, in the replication with NLTK neither of the claims of the original study can be confirmed.

Consider next the study of the sentiments grouped by the weekday. Guzman et al. report that comments on Monday were more negative than comments on the other days. Similarly to the study of programming languages, Table 14 suggests that a similar conclusion can be derived if SentiStrength is used but is no longer the case for NLTK. In fact, the mean NLTK score for Monday is the *least* negative. The median values both for SentiStrength and for NLTK are 0 for all the days suggesting no difference can be found. Then Guzman et al. have performed a statistical analysis and compared Monday against each of the other days. This analysis "confirmed that commit comments were more negative on Monday than on Sunday, Tuesday, and Wednesday (p-value ≤ 0.015). We replicated this study with SentiStrength and observed that $p \leq 0.015$ for Tuesday, Friday and Saturday. We can conclude that while the exact days have not been confirmed, at least we still can say that commit comments on Monday are more negative than those on *some other days*. Unfortunately, even a weaker conclusion cannot be confirmed if NLTK has been used: p exceeds the 0.015 for all days (in fact, $p \geq 0.72$ for all days).

Finally, Table 15 shows that NLTK evaluates the comments made in the afternoon as slightly more negative than comments in the evening, in contrast to SentiStrength that indicates the afternoon comments as the most positive, or at least the least negative ones. We could not replicate those results neither for SentiStrength nor for NLTK.

5.4 Discussion

When replicating the study of Pletea et al. we confirm the original observation that security comments or discussions are more often negative than the non-security comments or discussions. We also observe that the when compared with the manually labeled security discussions both tools produce mixed results. However, we could not find evidence supporting the suggestion that security-related discussions are more emotional.

When trying to replicate the results of Guzman et al. we could not derive the same conclusion when a different tool has been used. The only conclusion we could replicate when the same tool has been used is that the commit comments on Monday are more negative than those on *some other days*, which is a weakened form of the original claim. Recently Islam and Zibran (2016) have performed a similar study of the differences between emotions expressed by developers during different times and days of a week. Similarly to Guzman et al. Islam and Zibran have studied commit messages and used SentiStrength; as opposed Guzman et al. they have considered 50 projects with the highest number of commits from the Boa dataset (Dyer et al. 2013) rather than the 2014 MSR mining challenge dataset of 90 GitHub projects (Gousios 2013). In sharp contrast with the work of Guzman et al. no significant differences have been found in the developers' emotions in different times and days of a week.

Our replication studies show that validity of conclusions of the previously published papers such as the ones by Pletea et al. (2014) and Guzman et al. (2014) should be questioned and ideally reassessed when (or if) a sentiment analysis tool will become available specifically targeting software engineering domain.

5.5 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 the pairwise Wilcoxon test as done in Section 5.3.2 might not be the preferred approach from the statistical point of view: this is why a more advanced statistical technique has been used in Section 4. However, to support the comparative aspects of replication in Section 5.3.2 we present the results exactly in the same way as in the original work (Guzman et al. 2014).

6 Related Work

This paper builds on our previous work (Jongeling et al. 2015). The current submission extends it by reporting on a follow-up study (Section 3.3), replication of two recent studies (Section 5) as well presenting a more elaborate discussion of the related work below.

6.1 Sentiment Analysis in Large Text Corpora

As announced in the *Manifesto for Agile Software Development* (Beck et al. 2001), the centrality of developer interaction in large scale software development has come to be increasingly recognized in recent times (Datta et al. 2012; Schröter et al. 2012). 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 (Cataldo and Herbsleb 2008), developer interaction is facilitated by collaborative development environments that capture details of discussion around development activities (Costa et al. 2011). 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 (Pang et al. 2002). 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 (Davidov et al. 2010). 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 (Pak and Paroubek 2010) and Agarwal et al., have proposed an approach to correctly identify the polarity of tweets (Agarwal et al. 2011). Mohammad, Kiritchenko, and Xiaodan report the utility of using support vector machine (SVM) base classifiers while analyzing sentiments in tweets (Mohammad et al. 2013). Online question and answer forums such as Yahoo! Answers are also helpful sources for sentiment mining data (Kucuktunc et al. 2012).

6.2 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 (Goul et al. 2012). The authors conclude that while sentiment analysis can facilitate requirements engineering, in some cases algorithmic analysis of reviews can be problematic (Goul et al. 2012). 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 (Leopairote et al. 2013). 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 (Tourani et al. 2014). 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 (Novielli et al. 2015). Impact of the sentiment on issue resolution time, similar to *RQ3* discussed in Section 4, have also been considered in the literature (Garcia et al. 2013; Ortu et al. 2015).

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 (Vivian et al. 2015).

Latest studies have also explored the symbiotic relationship between collaborative software engineering and different kinds of task based emotions (Dewan 2015).

6.3 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 (Garcia et al. 2013; Guzman et al. 2014; Guzman and Bruegge 2013; Novielli et al. 2015; Ortu et al. 2015; Panichella et al. 2015; Pletea et al. 2014; Rousinopoulos et al. 2014)

With the notable exception of the work of Panichella et al. (2015) 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 (Guzman et al. 2014; Pletea et al. 2014).

In our previous work (Jongeling et al. 2015) 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. (2015) 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 (Barkmann et al. 2009) and code smell detection tools (Fontana et al. 2011). Similarly to our findings, disagreement between the tools was observed.

6.4 Replications and Negative Results

This paper builds on our previous work (Jongeling et al. 2015). The current submission extends it by reporting on replication of two recent studies (Section 5). 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" (Tonella et al. 2007). The challenges around replication studies in empirical software engineering have been identified by Mende (2010). de Magalhães 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 (de Magalhães et al. 2014). Recent studies have begun to address this replication gap (Sfetsos et al. 2012; Greiler et al. 2015).

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 (Pritchard 1984; Fuhr and Muller 1987), its importance is being reiterated in recent years (Giraud-Carrier and Dunham 2011). By carefully and objectively examining what went wrong in the quest for expected outcome, the state-of-art and practice can be enhanced (Lindsey 2011; Täht 2014). We believe the results reported in this paper can aid such enhancement.

7 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 diverging 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.

Acknowledgments 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.

Appendix A: Agreement of Sentiment Analysis Tools with the Manual Labeling and with each other

Table 16 presents the confusion matrices corresponding to Table 2. Similarly, Table 17 presents the confusion matrices corresponding to Table 3.

Table 16 Confusion matrices corresponding to Table 2

⇓	pos	neu	neg	⇓	pos	neu	neg
NLTK	Manual			SentiStrength	Manual		
pos	26	27	4	pos	30	53	3
neu	6	128	1	neu	10	126	7
neg	9	47	14	neg	1	23	9
Alchemy	Manual			Stanford NLP	Manual		
pos	31	39	3	pos	20	13	1
neu	3	74	1	neu	11	67	1
neg	7	89	15	neg	10	122	17
NLTK	SentiStrength			NLTK	Alchemy		
pos	32	21	4	pos	39	6	12
neu	34	89	12	neu	21	55	59
neg	20	33	17	neg	13	17	40
NLTK	Stanford NLP			SentiStrength	Alchemy		
pos	19	16	22	pos	44	13	29
neu	9	51	75	neu	26	62	55
neg	6	12	52	neg	3	3	27
SentiStrength	Stanford NLP			Alchemy	Stanford NLP		
pos	20	22	44	pos	23	16	34
neu	13	57	73	neu	6	32	40
neg	1	0	32	neg	5	31	75

Table 17 Confusion matrices corresponding to Table 3

NLTK and SentiStrength	Manual pos	neu	neg	NLTK and , Alchemy	Manual pos	neu	neg
pos	23	8	1	pos	23	14	2
neu	4	85	0	neu	2	53	0
neg	0	10	7	neg	3	24	13
NLTK and	Manual			Alchemy and	Manual		
Stanford NLP	pos	neu	neg	SentiStrength,	pos	neu	neg
pos	16	3	0	pos	26	17	1
neu	3	48	0	neu	2	59	1
neg	5	34	13	neg	1	18	8
SentiStrength	Manual			Alchemy	Manual		
and Stanford NLP	pos	neu	neg	and Stanford NLP	pos	neu	neg
pos	17	3	0	pos	19	4	0
neu	3	53	1	neu	1	30	1
neg	1	23	8	neg	5	56	14
NLTK, Alchemy	Manual			NLTK, Stanford NLP	Manual		
and SentiStrength	pos	neu	neg	and SentiStrength	pos	neu	neg
pos	21	5	1	pos	15	1	0
neu	2	43	0	neu	2	37	0
neg	0	9	7	neg	0	10	6
Alchemy, Stanford NLP	Manual			NLTK, Alchemy	Manual		
and SentiStrength	pos	neu	neg	and Stanford NLP	pos	neu	neg
pos	16	1	0	pos	15	2	0
neu	1	29	1	neu	1	23	0
neg	1	18	7	neg	3	19	12
all tools	Manual						
	pos	neu	neg				
pos	14	1	0				
neu	1	22	0				
neg	0	9	6				

Appendix B: Comparison of NLTK and SentiStrength in Combination with Politeness

Tables 18, 19 and 20 are similar to Table 7 and are provided for the sake of completeness.

Table 18 Comparison of NLTK and SentiStrength in combination with politeness for the Gnome dataset. Thresholds for statistical significance: 0.05 (*), 0.01 (**), 0.001 (***). Exact p-values are indicated as subscripts. 0 indicates that the p-value is too small to be computed precisely

	NLTK			SentiStrength			NLTK ∩ SentiStrength		
descr									
	neg	neu	pos	neg	neu	pos	neg	neu	pos
imp	43702	260570	15306	48835	259271	11472	14105	219444	1111
neu	9945	30794	4883	9513	33227	2882	2627	22958	617
pol	385	542	191	237	728	153	97	378	57
	neg.imp > neu.imp$_0^{***}$			neg.imp > neu.imp$_0^{***}$			neg.imp > neu.imp$_0^{***}$		

Table 18 (continued)

NLTK	SentiStrength	NLTK ∩ SentiStrength
$\text{neg.neu} > \text{neg.imp}_{0}^{***}$	$\text{neg.neu} > \text{neg.imp}_{0}^{***}$	
$\text{neg.neu} > \text{neu.imp}_{0}^{***}$	$\text{neg.neu} > \text{neu.imp}_{0}^{***}$	$\text{neg.neu} > \text{neu.imp}_{0}^{***}$
$\text{neg.neu} > \text{pos.imp}_{1.59\times10^{-3}}^{**}$	$\text{neg.neu} > \text{pos.imp}_{0}^{***}$	
$\text{neg.pol} > \text{neu.imp}_{1.62\times10^{-8}}^{***}$	$\text{neg.pol} > \text{neu.imp}_{9.54\times10^{-14}}^{***}$	$\text{neg.pol} > \text{neu.imp}_{2.16\times10^{-3}}^{**}$
	$\text{neg.pol} > \text{pos.imp}_{5.23\times10^{-4}}^{***}$	
$\text{neu.neu} > \text{neg.imp}_{0}^{***}$	$\text{neu.neu} > \text{neg.imp}_{0}^{***}$	$\text{neu.neu} > \text{neg.imp}_{1.16\times10^{-2}}^{*}$
$\text{neu.neu} > \text{neg.neu}_{1.65\times10^{-3}}^{**}$		
	$\text{neg.neu} > \text{neu.neu}_{6.78\times10^{-8}}^{***}$	
$\text{neu.neu} > \text{neu.imp}_{0}^{***}$	$\text{neu.neu} > \text{neu.imp}_{0}^{***}$	$\text{neu.neu} > \text{neu.imp}_{0}^{***}$
$\text{neu.neu} > \text{pos.imp}_{0}^{***}$	$\text{neu.neu} > \text{pos.imp}_{0}^{***}$	
$\text{neu.pol} > \text{neg.imp}_{1.59\times10^{-5}}^{***}$		
$\text{neu.pol} > \text{neu.imp}_{0}^{***}$	$\text{neu.pol} > \text{neu.imp}_{0}^{***}$	$\text{neu.pol} > \text{neu.imp}_{0}^{***}$
	$\text{neu.pol} > \text{pos.imp}_{4.95\times10^{-5}}^{***}$	
$\text{pos.imp} > \text{neg.imp}_{0}^{***}$	$\text{neg.imp} > \text{pos.imp}_{0}^{***}$	
$\text{pos.imp} > \text{neu.imp}_{0}^{***}$	$\text{pos.imp} > \text{neu.imp}_{0}^{***}$	$\text{pos.imp} > \text{neu.imp}_{0}^{***}$
$\text{pos.neu} > \text{neg.imp}_{0}^{***}$	$\text{pos.neu} > \text{neg.imp}_{1.9\times10^{-7}}^{***}$	$\text{pos.neu} > \text{neg.imp}_{3.29\times10^{-2}}^{*}$
$\text{pos.neu} > \text{neg.neu}_{1.6\times10^{-7}}^{***}$		
$\text{pos.neu} > \text{neg.pol}_{1.35\times10^{-2}}^{*}$		
$\text{pos.neu} > \text{neu.imp}_{0}^{***}$	$\text{pos.neu} > \text{neu.imp}_{0}^{***}$	$\text{pos.neu} > \text{neu.imp}_{0}^{***}$
$\text{pos.neu} > \text{neu.neu}_{1.54\times10^{-2}}^{*}$		
$\text{pos.neu} > \text{pos.imp}_{0}^{***}$	$\text{pos.neu} > \text{pos.imp}_{0}^{***}$	
$\text{pos.pol} > \text{neg.imp}_{5.29\times10^{-4}}^{***}$		
$\text{pos.pol} > \text{neu.imp}_{2.22\times10^{-16}}^{***}$	$\text{pos.pol} > \text{neu.imp}_{2.34\times10^{-6}}^{***}$	$\text{pos.pol} > \text{neu.imp}_{5.2\times10^{-5}}^{***}$

Table 19 Comparison of NLTK and SentiStrength in combination with politeness for the Stack Overflow datasets. Thresholds for statistical significance: 0.05 (*). 0.01 (**), 0.001 (***). Exact p-values are indicated as subscripts. 0 indicates that the p-value is too small to be computed precisely

	NLTK			SentiStrength			NLTK ∩ SentiStrength		
title									
	neg	neu	pos	neg	neu	pos	neg	neu	pos
imp	61	244	29	43	270	21	11	203	5
neu	19	37	12	10	55	3	5	34	3
pol	4	4	0	0	5	3	0	3	0
							$\text{neutral.polite} > \text{pos.impolite}_{0}^{***}$		
descr									
	neg	neu	pos	neg	neu	pos	neg	neu	pos
imp	33	7	4	12	24	8	11	4	0

Table 19 (continued)

	NLTK			SentiStrength			NLTK ∩ SentiStrength		
neu	38	20	9	15	32	20	10	8	2
pol	178	44	77	63	127	109	41	23	40
	neg.neutral > pos.impolite$^{***}_{2,37\times10^{-4}}$								
	neg.polite > pos.impolite$^{*}_{4,87\times10^{-2}}$								
	pos.polite > pos.impolite$^{**}_{5,82\times10^{-3}}$								

Table 20 Comparison of NLTK and SentiStrength in combination with politeness for the ASF datasets. Thresholds for statistical significance: 0.05 (*). 0.01 (**), 0.001 (***). Exact *p*-values are indicated as subscripts. 0 indicates that the p-value is too small to be computed precisely

	NLTK			SentiStrength			NLTK ∩ SentiStrength		
title									
	neg	neu	pos	neg	neu	pos	neg	neu	pos
imp	15690	55726	5819	19228	50437	7573	5216	37083	733
neu	3527	11988	2404	4799	11265	1856	1195	7583	340
pol	150	234	125	114	314	81	39	152	33
	neg.imp > neg.neu$^{**}_{6.51\times10^{-3}}$								
				neg.imp > neu.neu$^{**}_{6.05\times10^{-3}}$					
	neu.imp > neg.neu$^{**}_{5.97\times10^{-3}}$								
	neu.neu > neg.neu$^{*}_{1.29\times10^{-2}}$								
				neg.neu > neu.neu$^{*}_{2.9\times10^{-2}}$					
				pos.imp > neg.imp$^{***}_{1.55\times10^{-10}}$					
	pos.imp > neg.neu$^{***}_{8.81\times10^{-4}}$			pos.imp > neg.neu$^{***}_{7.53\times10^{-4}}$					
				pos.imp > neu.imp$^{***}_{0}$					
				pos.imp > neu.neu$^{***}_{0}$					
				pos.neu > neg.imp$^{*}_{1.73\times10^{-2}}$					
	pos.neu > neg.neu$^{*}_{3.14\times10^{-2}}$								
				pos.neu > neu.imp$^{***}_{3.04\times10^{-4}}$					
				pos.neu > neu.neu$^{***}_{6.62\times10^{-6}}$					
descr									
	neg	neu	pos	neg	neu	pos	neg	neu	pos
imp	5293	10291	1881	5553	9595	2346	1937	5816	358
neu	9505	16709	4357	10357	15205	5008	3501	8425	1048
pol	15493	15433	6872	13041	16161	8586	5530	6646	2401
	a			*b*			*c*		
	neg.imp > neu.imp$^{**}_{1.06\times10^{-3}}$								
	neg.neu > neg.imp$^{*}_{2.92\times10^{-2}}$						neg.neu > neg.imp$^{*}_{3.36\times10^{-2}}$		
	neg.neu > neu.imp$^{***}_{0}$			neg.neu > neu.imp$^{***}_{6.23\times10^{-6}}$			neg.neu > neu.imp$^{***}_{7.57\times10^{-14}}$		
	neg.neu > neu.neu$^{***}_{9.43\times10^{-7}}$						neg.neu > neu.neu$^{***}_{4.84\times10^{-7}}$		
	neg.pol > neg.imp$^{***}_{0}$			neg.pol > neg.imp$^{***}_{0}$			neg.pol > neg.imp$^{***}_{0}$		
	neg.pol > neg.neu$^{***}_{0}$			neg.pol > neg.neu$^{***}_{0}$			neg.pol > neg.neu$^{***}_{0}$		
	neg.pol > neu.imp$^{***}_{0}$			neg.pol > neu.imp$^{***}_{0}$			neg.pol > neu.imp$^{***}_{0}$		

Table 20 (continued)

NLTK	SentiStrength	NLTK ∩ SentiStrength
$\text{neg.pol} > \text{neu.neu}_{0}^{***}$		$\text{neg.pol} > \text{neu.neu}_{0}^{***}$
	$\text{neu.pol} > \text{neg.pol}_{2.49\times10^{-3}}^{**}$	
$\text{neg.pol} > \text{neu.pol}_{0}^{***}$	$\text{neu.pol} > \text{neg.pol}_{2.49\times10^{-3}}^{**}$	
$\text{neg.pol} > \text{pos.imp}_{0}^{***}$	$\text{neg.pol} > \text{pos.imp}_{4.56\times10^{-10}}^{***}$	
$\text{neg.pol} > \text{pos.neu}_{0}^{***}$	$\text{neg.pol} > \text{pos.neu}_{8.89\times10^{-6}}^{***}$	
$\text{neu.neu} > \text{neu.imp}_{2.83\times10^{-5}}^{***}$	$\text{neu.neu} > \text{neu.imp}_{2.34\times10^{-2}}^{*}$	$\text{neu.neu} > \text{neu.imp}_{1.53\times10^{-2}}^{*}$
$\text{neu.pol} > \text{neg.imp}_{0}^{***}$	$\text{neu.pol} > \text{neg.imp}_{0}^{***}$	$\text{neu.pol} > \text{neg.imp}_{0}^{***}$
$\text{neu.pol} > \text{neg.neu}_{0}^{***}$	$\text{neu.pol} > \text{neg.neu}_{0}^{***}$	$\text{neu.pol} > \text{neg.neu}_{6.2\times10^{-13}}^{***}$
$\text{neu.pol} > \text{neu.imp}_{0}^{***}$	$\text{neu.pol} > \text{neu.imp}_{0}^{***}$	$\text{neu.pol} > \text{neu.imp}_{0}^{***}$
$\text{neu.pol} > \text{neu.neu}_{0}^{***}$	$\text{neu.pol} > \text{neu.neu}_{0}^{***}$	$\text{neu.pol} > \text{neu.neu}_{0}^{***}$
$\text{neu.pol} > \text{pos.imp}_{2.79\times10^{-9}}^{***}$	$\text{neu.pol} > \text{pos.imp}_{0}^{***}$	
$\text{neu.pol} > \text{pos.neu}_{3.99\times10^{-14}}^{***}$	$\text{neu.pol} > \text{pos.neu}_{7.07\times10^{-14}}^{***}$	
	$\text{pos.imp} > \text{neg.imp}_{1.91\times10^{-3}}^{**}$	
$\text{pos.imp} > \text{neu.imp}_{1.82\times10^{-4}}^{***}$	$\text{pos.imp} > \text{neu.imp}_{2.06\times10^{-6}}^{***}$	$\text{pos.imp} > \text{neu.imp}_{2.89\times10^{-3}}^{**}$
	$\text{pos.imp} > \text{neu.neu}_{1.38\times10^{-2}}^{*}$	
$\text{pos.neu} > \text{neg.imp}_{2.06\times10^{-2}}^{*}$	$\text{pos.neu} > \text{neg.imp}_{0}^{***}$	$\text{pos.neu} > \text{neg.imp}_{2.03\times10^{-9}}^{***}$
	$\text{pos.neu} > \text{neg.neu}_{1.84\times10^{-13}}^{***}$	$\text{pos.neu} > \text{neg.neu}_{3.49\times10^{-4}}^{***}$
$\text{pos.neu} > \text{neu.imp}_{2.24\times10^{-13}}^{***}$	$\text{pos.neu} > \text{neu.imp}_{0}^{***}$	$\text{pos.neu} > \text{neu.imp}_{0}^{***}$
$\text{pos.neu} > \text{neu.neu}_{1.7\times10^{-5}}^{***}$	$\text{pos.neu} > \text{neu.neu}_{0}^{***}$	$\text{pos.neu} > \text{neu.neu}_{8.22\times10^{-15}}^{***}$
$\text{pos.pol} > \text{neg.imp}_{0}^{***}$	$\text{pos.pol} > \text{neg.imp}_{0}^{***}$	$\text{pos.pol} > \text{neg.imp}_{0}^{***}$
$\text{pos.pol} > \text{neg.neu}_{0}^{***}$	$\text{pos.pol} > \text{neg.neu}_{0}^{***}$	$\text{pos.pol} > \text{neg.neu}_{0}^{***}$
	$\text{pos.pol} > \text{neg.pol}_{2.45\times10^{-12}}^{***}$	$\text{pos.pol} > \text{neg.pol}_{4.21\times10^{-2}}^{*}$
$\text{pos.pol} > \text{neu.imp}_{0}^{***}$	$\text{pos.pol} > \text{neu.imp}_{0}^{***}$	$\text{pos.pol} > \text{neu.imp}_{0}^{***}$
$\text{pos.pol} > \text{neu.neu}_{0}^{***}$	$\text{pos.pol} > \text{neu.neu}_{0}^{***}$	$\text{pos.pol} > \text{neu.neu}_{0}^{***}$
$\text{pos.pol} > \text{neu.pol}_{1.54\times10^{-12}}^{***}$	$\text{pos.pol} > \text{neu.pol}_{1.24\times10^{-3}}^{**}$	$\text{pos.pol} > \text{neu.pol}_{1.79\times10^{-6}}^{***}$
$\text{pos.pol} > \text{pos.imp}_{0}^{***}$	$\text{pos.pol} > \text{pos.imp}_{0}^{***}$	$\text{pos.pol} > \text{pos.imp}_{1.57\times10^{-2}}^{*}$
$\text{pos.pol} > \text{pos.neu}_{0}^{***}$	$\text{pos.pol} > \text{pos.neu}_{0}^{***}$	$\text{pos.pol} > \text{pos.neu}_{3.06\times10^{-2}}^{*}$

[a] Sentiment of 174 descriptions could not been determined.

[b] Sentiment of 183 descriptions could not been determined.

[c] Sentiment of 81 descriptions could not been determined.

References

Abbasi A, Hassan A, Dhar M (2014) Benchmarking Twitter sentiment analysis tools. In: International Conference on Language Resources and Evaluation. ELRA, Reykjavik, Iceland, pp 823–829

Agarwal A, Xie B, Vovsha I, Rambow O, Passonneau R (2011) 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. http://dl.acm.org/citation.cfm?id=2021109.2021114

Asaduzzaman M, Bullock MC, Roy CK, Schneider KA Bug introducing changes: A case study with android. In: Lanza et al. [43], pp 116–119. doi:10.1109/MSR.2012.6224267

Baccianella S, Esuli A, Sebastiani F (2010) SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta. http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf

Bakeman R, Gottman JM (1997) Observing interaction: an introduction to sequential analysis. Cambridge University Press. https://books.google.nl/books?id=CMj2SmcijhEC

Barkmann H, Lincke R, Löwe W (2009) 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

Batista GEAPA, Carvalho ACPLF, Monard MC (2000) Applying one-sided selection to unbalanced datasets. In: Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence. Springer-Verlag, London, UK, UK, pp 315–325

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 RC, Mellor S, Schwaber K, Sutherland J, Thomas D (2001) Manifesto for agile software development http://agilemanifesto.org/principles.html Last accessed: October 14, 2015

Bird S, Loper E, Klein E (2009) Natural language processing with Python. O'Reilly Media Inc

Brunner E, Munzel U (2000) The Nonparametric Behrens-Fisher Problem: Asymptotic Theory and a Small-Sample Approximation. Biometrical Journal 42(1):17–25

Capiluppi A, Serebrenik A, Singer L (2013) Assessing technical candidates on the social web. Software. IEEE 30(1):45–51. doi:10.1109/MS.2012.169

Cataldo M, Herbsleb JD (2008) Communication networks in geographically distributed software development. In: Proceedings of the 2008 ACM conference on Computer supported cooperative work, CSCW '08, pp 579–588. ACM, New York, NY, USA. doi:10.1145/1460563.1460654

Cohen J (1968) Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. Psychol Bull 70(4):213–220

Costa JM, Cataldo M, de Souza CR (2011) 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, pp 3151–3160. ACM, New York, NY, USA. doi:10.1145/1978942.1979409

Dajsuren Y, van den Brand MGJ, Serebrenik A, Roubtsov S (2013) Simulink models are also software: Modularity assessment. In: Proceedings of the 9th International ACM Sigsoft Conference on Quality of Software Architectures, QoSA '13, pp 99–106. ACM, New York, NY, USA. doi:10.1145/2465478.2465482

Danescu-Niculescu-Mizil C, Sudhof M, Jurafsky D, Leskovec J, Potts C (2013) A computational approach to politeness with application to social factors. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers, pp 250–259. The Association for Computer Linguistics. http://aclweb.org/anthology/P/P13/P13-1025.pdf

Datta S, Sindhgatta R, Sengupta B (2012) 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. doi:10.1145/2384616.2384664

Davidov D, Tsur O, Rappoport A (2010) 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. http://dl.acm.org/citation.cfm?id=1870568.1870582

de Magalhães CVC, da Silva FQB, Santos RES (2014) 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. doi:10.1145/2601248.2601289

Destefanis G, Ortu M, Counsell S, Swift S, Marchesi M, Tonelli R (2016) Peer J Comput Sci 2(e73):1–35. doi:10.7717/peerj-cs.73

Dewan P (2015) 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. doi:10.1109/CHASE.2015.32

Dunn OJ (1961) Multiple comparisons among means. J Am Stat Assoc 56(293):52–64

Dyer R, Nguyen HA, Rajan H, Nguyen TN (2013) Boa: A language and infrastructure for analyzing ultra-large-scale software repositories. In: Proceedings of the 2013 International Conference on Software Engineering, ICSE '13, pp 422–431. IEEE Press, Piscataway, NJ, USA. http://dl.acm.org/citation.cfm?id=2486788.2486844

Fleiss JL, Levin B, Paik MC (2003) Statistical methods for rates and proportions, 3rd edn. Wiley Series in Probability and Statistics. Wiley, Hoboken, NJ

Fontana FA, Mariani E, Morniroli A, Sormani R, Tonello A (2011) An experience report on using code smells detection tools. In: ICST Workshops, pp 450–457. IEEE

Fuhr N, Muller P (1987) 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. doi:10.1145/42005.42007

Gabriel KR (1969) Simultaneous test procedures—some theory of multiple comparisons. Ann Math Stat 40(1):224–250

Gamon M, Aue A, Corston-Oliver S, Ringger E (2005) Pulse: Mining customer opinions from free text. In: Proceedings of the 6th International Conference on Advances in Intelligent Data Analysis, IDA'05. Springer-Verlag, Berlin, Heidelberg, pp 121–132. doi:10.1007/11552253_12

Garcia D, Zanetti MS, Schweitzer F (2013) 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

Giraud-Carrier C, Dunham MH (2011) On the importance of sharing negative results. Sigkdd Explor. Newsl. 12(2):3–4. doi:10.1145/1964897.1964899

Goul M, Marjanovic O, Baxley S, Vizecky K (2012) 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. doi:10.1109/HICSS.2012.421

Gousios G (2013) The GHTorrent dataset and tool suite. In: Proceedings of the 10th Working Conference on Mining Software Repositories, MSR'13, pp 233–236. http://dl.acm.org/citation.cfm?id=2487085.2487132

Greiler M, Herzig K, Czerwonka J (2015) 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. http://dl.acm.org.library.sutd.edu.sg:2048/citation.cfm?id=2820518.2820522

Guzman E, Azócar D, Li Y (2014) Sentiment analysis of commit comments in GitHub: An empirical study. In: MSR, pp 352–355, ACM, New York, NY, USA

Guzman E, Bruegge B (2013) Towards emotional awareness in software development teams. In: Joint Meeting on Foundations of Software Engineering, pp 671–674, ACM, New York, NY, USA

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The Weka data mining software: An upyear. SIGKDD Explor Newsl 11(1):10–18

Honkela T, Izzatdust Z, Lagus K (2012) Text mining for wellbeing: Selecting stories using semantic and pragmatic features. In: Artificial Neural Networks and Machine Learning, Part II, LNCS, vol 7553. Springer, pp 467–474

Howard MJ, Gupta S, Pollock LL, Vijay-Shanker K (2013) Automatically mining software-based, semantically-similar words from comment-code mappings. In: Zimmermann T, Penta MD, Kim S (eds) MSR, pp 377–386. IEEE Computer Society

Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218. doi:10.1007/BF01908075

Islam MR, Zibran MF (2016) Towards understanding and exploiting developers' emotional variations in software engineering. In: 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA), pp 185–192. doi:10.1109/SERA.2016.7516145

Jongeling R, Datta S, Serebrenik A (2015) Choosing your weapons: On sentiment analysis tools for software engineering research. In: ICSME, pp 531–535. IEEE. doi:10.1109/ICSM.2015.7332508

Konietschke F, Hothorn LA, Brunner E (2012) Rank-based multiple test procedures and simultaneous confidence intervals. Electronic Journal of Statistics 6:738–759

Kucuktunc O, Cambazoglu BB, Weber I, Ferhatosmanoglu H (2012) 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. doi:10.1145/2124295.2124371

Lanza M, Di Penta M, Xie T (2012) (eds.): 9th IEEE Working Conference of Mining Software Repositories, MSR 2012, June 2-3, 2012, Zurich, Switzerland. IEEE Computer Society. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6220358

Leopairote W, Surarerks A, Prompoon N (2013) 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. doi:10.1109/JCSSE.2013.6567355

Lewis DD, Gale WA (1994) A sequential algorithm for training text classifiers. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '94, pp. 3–12. Springer-Verlag New York, Inc., New York, NY, USA. http://dl.acm.org.dianus.libr.tue.nl/citation.cfm?id=188490.188495

Li TH, Liu R, Sukaviriya N, Li Y, Yang J, Sandin M, Lee J (2014) Incident ticket analytics for it application management services. In: 2014 IEEE International Conference on Services Computing (SCC), pp 568–574. doi:10.1109/SCC.2014.80

Lindsey MR (2011) 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. doi:10.1145/2124436.2124453
Linstead E, Baldi P (2009) Mining the coherence of GNOME bug reports with statistical topic models. In: Godfrey MW, Whitehead J (eds) Proceedings of the 6th International Working Conference on Mining Software Repositories, MSR 2009 (Co-located with ICSE), Vancouver, BC, Canada, May 16–17, 2009, Proceedings, pp 99–102. IEEE Computer Society. doi:10.1109/MSR.2009.5069486
Martie L, Palepu VK, Sajnani H, Lopes CV Trendy bugs: Topic trends in the android bug reports. In: Lanza et al. [43], pp 120–123. doi:10.1109/MSR.2012.6224268
Mende T (2010) 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. doi:10.1145/1868328.1868336
Mishne G, Glance NS (2006) Predicting movie sales from blogger sentiment. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, pp 155–158
Mohammad SM, Kiritchenko S, Zhu X (2013) NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. arXiv:1308.6242[cs]
Murgia A, Tourani P, Adams B, Ortu M (2014) Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In: MSR, pp 262-271, ACM, New York, NY, USA
Novielli N, Calefato F, Lanubile F (2015) 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. doi:10.1145/2804381.2804387
Ortu M, Adams B, Destefanis G, Tourani P, Marchesi M, Tonelli R (2015) Are bullies more productive? empirical study of affectiveness vs. issue fixing time. In: MSR
Ortu M, Destefanis G, Adams B, Murgia A, Marchesi M, Tonelli R (2015) The JIRA repository dataset: Understanding social aspects of software development. In: Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering, PROMISE '15, pp 1:1–1:4. ACM, New York, NY, USA. doi:10.1145/2810146.2810147
Ortu M, Destefanis G, Kassab M, Counsell S, Marchesi M, Tonelli R (2015) Would you mind fixing this issue? - an empirical analysis of politeness and attractiveness in software developed using agile boards. In: Lassenius C, Dingsøyr T, Paasivaara M (eds) Agile Processes, in Software Engineering, and Extreme Programming - 16th International Conference, XP 2015, Helsinki, Finland, May 25–29, 2015, Proceedings, Lecture Notes in Business Information Processing, vol 212. Springer, pp 129–140. doi:10.1007/978-3-319-18612-2_11
Pak A, Paroubek P (2010) 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. http://dl.acm.org/citation.cfm?id=1859664.1859761
Pang B, Lee L (2007) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1-2):1–135
Pang B, Lee L, Vaithyanathan S (2002) 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. doi:10.3115/1118693.1118704
Panichella S, Sorbo AD, Guzman E, Visaggio CA, Canfora G, Gall HC (2015) How can I improve my app? classifying user reviews for software maintenance and evolution. In: ICSME. IEEE, pp 281–290
Pletea D, Vasilescu B, Serebrenik A (2014) Security and emotion: Sentiment analysis of security discussions on GitHub. In: MSR. ACM, New York, NY, USA, pp 348-351. doi:10.1145/2597073.2597117
Pritchard P (1984) Some negative results concerning prime number generators. Commun ACM 27(1):53–57. doi:10.1145/69605.357970
Rand WM (1971) Objective criteria for the evaluation of clustering methods. J Am Stat Assoc 66(336):846–850
Rousinopoulos AI, Robles G, González-Barahona JM (2014) 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
Santos JM, Embrechts M (2009) 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. Springer, pp 175–184
Schröter A, Aranda J, Damian D, Kwan I (2012) 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, pp 1317–1326. ACM, New York, NY, USA. doi:10.1145/2145204.2145401
Sfetsos P, Adamidis P, Angelis L, Stamelos I, Deligiannis I (2012) Investigating the impact of personality and temperament traits on pair programming: a controlled experiment replication. In: 2012 Eighth International Conference on the Quality of Information and Communications Technology (QUATIC), pp 57–65. doi:10.1109/QUATIC.2012.36

Sheskin DJ (2007) Handbook of parametric and nonparametric statistical procedures, 4 edn. Chapman & Hall

Shihab E, Kamei Y, Bhattacharya P (2012) Mining challenge 2012: the Android platform. In: MSR, pp 112–115

Shull FJ, Carver JC, Vegas S, Juristo N (2008) The role of replications in empirical software engineering. Empir Softw Eng 13(2):211–218. doi:10.1007/s10664-008-9060-1

Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng A, Potts C (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: Empirical Methods in Natural Language Processing, pp 1631–1642. Ass. for Comp. Linguistics

Sun X, Li B, Leung H, Li B, Li Y (2015) MSR4SM: Using topic models to effectively mining software repositories for software maintenance tasks. Inf Softw Technol 66:1–12. doi:10.1016/j.infsof.2015.05.003. http://www.sciencedirect.com/science/article/pii/S0950584915001007

Täht D (2014) 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. doi:10.1145/2630088.2652480

Thelwall M, Buckley K, Paltoglou G (2012) Sentiment strength detection for the social web. J Am Soc Inf Sci Technol 63(1):163–173

Thelwall M, Buckley K, Paltoglou G, Cai D, Kappas A (2010) Sentiment in short strength detection informal text. J Am Soc Inf Sci Technol 61(12):2544–2558

Tonella P, Torchiano M, Du Bois B, Systä T (2007) Empirical studies in reverse engineering: State of the art and future trends. Empir Softw Eng 12(5):551–571. doi:10.1007/s10664-007-9037-5

Tourani P, Jiang Y, Adams B (2014) 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. http://dl.acm.org/citation.cfm?id=2735522.2735528

Tukey JW (1951) Quick and dirty methods in statistics, part II, Simple analysis for standard designs. In: American Society for Quality Control, pp 189–197

Tumasjan A, Sprenger TO, Sandner PG, Welpe IM (2010) Predicting elections with twitter: What 140 characters reveal about political sentiment. In: International AAAI Conference on Weblogs and Social Media, pp 178–185

van Rijsbergen CJ (1979) Information Retrieval, 2nd edn. Butterworth-Heinemann, Newton, MA, USA

Vasilescu B, Filkov V, Serebrenik A (2013) StackOverflow and GitHub: associations between software development and crowdsourced knowledge. In: 2013 International Conference on Social Computing (SocialCom), pp 188–195. doi:10.1109/SocialCom.2013.35

Vasilescu B, Serebrenik A, van den Brand MGJ (2011) By no means: a study on aggregating software metrics. In: Concas G, Tempero ED, Zhang H, Penta MD (eds) Proceedings of the 2nd International Workshop on Emerging Trends in Software Metrics, WETSoM 2011, Waikiki, Honolulu, HI, USA, May 24, 2011. ACM, pp 23–26. doi:10.1145/1985374.1985381

Vasilescu B, Serebrenik A, Goeminne M, Mens T (2013) On the variation and specialisation of workload – a case study of the Gnome ecosystem community. Empir Softw Eng 19(4):955–1008. doi:10.1007/s10664-013-9244-1

Viera AJ, Garrett JM (2005) Understanding interobserver agreement: the kappa statistic. Fam Med 37(5):360–363

Vivian R, Tarmazdi H, Falkner K, Falkner N, Szabo C (2015) 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. http://dl.acm.org/citation.cfm?id=2819009.2819070

Wang S, Lo D, Vasilescu B, Serebrenik A (2014) EnTagRec: An enhanced tag recommendation system for software information sites. In: ICSME. IEEE, pp 291–300

Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1(6):80–83

Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 347–354

Yu HF, Ho CH, Juan YC, Lin CJ (2013) Libshorttext: A library for short-text classification and analysis. Tech. rep., Technical Report. http://www.csie.ntu.edu.tw/~cjlin/papers/libshorttext.pdf

Yu Y, Wang H, Yin G, Wang T (2016) Reviewer recommendation for pull-requests in github: What can we learn from code review and bug assignment? Inf Softw Technol 74:204–218. doi: 10.1016/j.infsof.2016.01.004. http://www.sciencedirect.com/science/article/pii/S0950584916000069

Zimmerman DW, Zumbo BD (1992) Parametric alternatives to the Student t test under violation of normality and homogeneity of variance. Percept Mot Skills 74(31):835–844

Robbert Jongeling is a consultant at ALTEN Technology in the Netherlands. He received a MSc degree in Computer Science and Engineering from Eindhoven University of Technology. After graduation in March of 2016, he has started his career in software design and engineering. His research interests include empirical software engineering.

Proshanta Sarkar is an application developer at IBM India Pvt. Ltd.; He received the M.Tech degree in Computer science and Engineering from Heritage Institute of Technology, India. He has more than 2 years of experience in roles of application developer across several client engagements in the design, development, and deployment of large scale enterprise software systems. His research interests include empirical software engineering, cognitive computing and BlockChain.

Subhajit Datta is currently a lecturer at the Singapore University of Technology and Design. He has more than 17 years of experience in software design, development, research, and teaching at various organizations in the United States of America, India, and Singapore. He is the author of the books Software Engineering: Concepts and Applications (Oxford University Press, 2010) and Metrics- Driven Enterprise Software Development (J. Ross Publishing, 2007), which are widely used by students and practitioners. His research interests include software architecture, empirical software engineering, social computing, and big data. Subhajit received the PhD degree in computer science from the Florida State University. More details about his background and interest are available at www.dattas.net.

Alexander Serebrenik (PhD, K.U. Leuven, Belgium 2003; MSc, Hebrew University, Israel, 1999) is associate professor software evolution at Eindhoven University of Technology. He has co-authored a book "Evolving Software Systems" (Springer Verlag, 2014), more than 100 scientific papers and articles. He is and was the chair of the steering committee chair, general chair and program chair of several conferences in the area of software maintenance and evolution. His research pertains both to technical and social aspects of software evolution.