



Development and Application of Sentiment Analysis Tools in Software Engineering: A Systematic Literature Review

Martin Obaidi

Leibniz Universität Hannover

Software Engineering Group

Germany

martin.obaidi@inf.uni-hannover.de

Jil Klünder

Leibniz Universität Hannover

Software Engineering Group

Germany

jil.kluender@inf.uni-hannover.de

ABSTRACT

Software development is a collaborative task and, hence, involves different persons. Research has shown the relevance of social aspects in the development team for a successful and satisfying project closure. Especially the mood of a team has been proven to be of particular importance. Thus, project managers or project leaders want to be aware of situations in which negative mood is present to allow for interventions. So-called sentiment analysis tools offer a way to determine the mood based on text-based communication. In this paper, we present the results of a systematic literature review of sentiment analysis tools developed for or applied in the context of software engineering. Our results summarize insights from 80 papers with respect to (1) the application domain, (2) the purpose, (3) the used data sets, (4) the approaches for developing sentiment analysis tools and (5) the difficulties researchers face when applying sentiment analysis in the context of software projects. According to our results, sentiment analysis is frequently applied to open-source software projects, and most tools are based on support-vector machines. Despite the frequent use of sentiment analysis in software engineering, there are open issues, e.g., regarding the identification of irony or sarcasm, pointing to future research directions.

CCS CONCEPTS

• **Software and its engineering** → **Collaboration in software development**; • **Human-centered computing** → *Collaborative and social computing systems and tools*.

KEYWORDS

Social Software Engineering, Sentiment Analysis, Machine Learning, Systematic Literature Review

ACM Reference Format:

Martin Obaidi and Jil Klünder. 2021. Development and Application of Sentiment Analysis Tools in Software Engineering: A Systematic Literature Review. In *Evaluation and Assessment in Software Engineering (EASE 2021)*, June 21–23, 2021, Trondheim, Norway. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3463274.3463328>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EASE 2021, June 21–23, 2021, Trondheim, Norway

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-9053-8/21/06...\$15.00

<https://doi.org/10.1145/3463274.3463328>

1 INTRODUCTION

Due to their growing complexity, software projects are rarely handled by individual developers, but by a team of developers [38, 61]. Often, these teams are distributed which increases the need for coordination and, hence, for interaction. The HELENA study from 2017 on methods used in software systems development shows that about 60% of all development teams ($n = 1006$) are globally distributed [39]. In this context, digital communication tools or channels such as e-mails, Slack¹, or JIRA² are even more important than in co-located teams where a lot of communication takes place face-to-face [53, 67]. For meetings, Schneider et al. [65] have shown that negative mood of an individual in the meeting can quickly affect the whole team and afterwards everyone can be demotivated. However, sentiments do not only affect the relationship between two people, but also the productivity, task synchronization and job satisfaction [11, 20, 21, 24].

Therefore, detecting bad mood is a goal pursued by many researchers (cf. [3, 9, 10, 28, 30, 54]). So-called sentiment analysis tools offer a way to determine the mood based on text-based communication. Besides, sentiment analysis tools have also been applied in other application scenarios, including the development of improvement suggestions for codes or recommendations for better software packages and libraries [49, 50, 56, 60]. There is a number of sentiment analysis tools developed and applied in the context of software engineering (SE) [3, 8, 32]. However, several tools are better suited for different contexts [54] and have been applied for different reasons and in different scenarios. To get an overview of the state-of-research on sentiment analysis in SE, we conducted a systematic literature review (SLR).

In this paper, we investigate which sentiment analysis tools are used in what domains of the SE and what data sources are used. More concretely, we contribute a list of application scenarios of sentiment analysis tools ranging from applications in academia over applications in open-source projects to applications in industry with their specific purposes, the used data sources, the approaches used for classification and the problems encountered during the development of such tools.

Outline: The rest of the paper is structured as follows: In Section 2, we present related work. The design of the literature review and the research methodology are explained in Section 3. The results are presented in Section 4. In Section 5, we discuss our results, before concluding the paper in Section 6.

¹<https://slack.com/>

²<https://www.atlassian.com/de/software/jira>

2 RELATED WORK

Several authors analyzed the use of specific sentiment analysis tools in SE. Zhang et al. [79] compared sentiment analysis tools like Senti4SD [8] and SentiCR [3] with each other. In addition, they described models based on the neural network BERT [14], which were trained with data related to SE such as GitHub³ or Stack Overflow⁴. In their replication study, Novielli et al. [58] explained some sentiment analysis tools (e.g. Senti4SD [8]) in great detail and described the underlying data.

Similarly, other papers compared sentiment analysis tools in their accuracy and described them in terms of their operation [52, 54]. Other papers mentioned some tools, too, but only briefly described them without going into details [6, 10, 29]. In contrast to our work, the authors did not follow a systematic approach to consider the broad range of existing literature and tools, but rather focused on specific papers only. They did not go into detail about why they chose these tools or data and what tools are available.

Nevertheless, there are literature reviews in the field of sentiment analysis that are not related to SE. Kumar and Jaiswal [40] conducted a SLR with the goal of advancing the understanding of the feasibility, scope, and relevance of studies that apply soft computing techniques for sentiment analysis. They considered tools which used Twitter data and identified research gaps in the field. These gaps include an incessant need to enhance the performance of the sentiment classification tools and the usage of other data sets like Flickr⁵ or Tumblr⁶. Abo et al. [1] conducted a systematic mapping study dealing with sentiment analysis for Arabic texts in social media. Devika et al. [13] looked at different approaches to sentiment analysis. Among other approaches such as support-vector machine (SVM), Naive Bayes classifier, they explained rule-based as well as lexicon-based methods. Maitama et al. [47] performed a systematic mapping study, which contains an examination of aspect-based sentiment analysis tools and an investigation of their approach, technique, diversity and demography.

However, all these SLRs are not related to SE, and the data or tools are not designed to the domain of SE. Consequently, no information about areas or motivation to use the tools in the context of software development is offered.

3 LITERATURE REVIEW

In order to gain an overview of the current state of sentiment analysis in the context of SE, we conducted a literature review. In particular, we strive towards reaching the following goal formulated as proposed by Wohlin et al. [76]:

Research Goal:

Analyze existing literature for the purpose of identifying widely used sentiment analysis methods and tools with respect to different application scenarios in software engineering from the point of view of a researcher in the context of a literature review.

³<https://github.com/>

⁴<https://stackoverflow.com/>

⁵<https://www.flickr.com/>

⁶<https://www.tumblr.com/>

3.1 Research question

In order to achieve the research goal and to analyze the literature on sentiment analysis in software engineering from different viewpoints, we pose the following research questions:

RQ 1: *What are the main application scenarios for sentiment analysis in the context of SE?* As a first step, we want to get an overview of the broad area of possible application scenarios in which sentiment analysis is used in the context of software projects.

RQ 2: *For what purpose is sentiment analysis used in the investigated studies?* Next, we want to analyze the different reasons why sentiment analysis is performed.

RQ 3: *What data is used as a basis for sentiment analysis?* We want to get an overview of the data used to train or evaluate the tools. This way, we investigate which data is suitable as a basis for sentiment analysis in development teams – both as training and/or test data.

RQ 4: *Which approaches are used when developing sentiment analysis tools?* With this question, we gain an overview of good practices in the application of sentiment analysis in SE.

RQ 5: *What are the difficulties of these approaches?* Last, we analyze the advantages and disadvantages of existing tools, problems and difficulties, etc. These insights point to future research directions that should be investigated to improve the applicability and the outcome of sentiment analysis tools.

3.2 Method

To provide an overview of the development and application of sentiment analysis in the context of SE, we conducted a SLR. Our approach is based on the research process proposed by Kitchenham et al. [5, 35] as well as Petersen et al. [62] and comprises five steps which we describe in the subsequent sections.

3.2.1 Selection of databases. While some papers are available via several scientific databases, others are not. Therefore, we included a total of five databases in our search to reduce the risk of missing papers which are only available in one database. Our selection comprises databases that are often used in SLRs in the SE domain [18, 36, 37, 64, 66, 77, 78]: Science Direct⁷, IEEE Xplore⁸, ACM Digital Library⁹, Springer Link¹⁰, and Google Scholar¹¹. We conducted a comprehensive search as proposed by Petersen et al. [62] in each of these databases in order to reduce biases.

3.2.2 Definition of the search string. The search string comprises keywords related to our research questions and basically, the search string uses terms of the two fields of *sentiment analysis* and *SE*. As we focus on SE at its core, we also added related terms such as “software project” (which is the typical use case of sentiment analysis) and “development team” (which is most often the object whose sentiments are studied). We also extracted synonyms for sentiment analysis like “opinion mining” from different papers [12, 43, 44] and considered them in addition to the term “sentiment analysis”. Finally, we obtained the following search string:

⁷<https://www.sciencedirect.com/>

⁸<https://ieeexplore.ieee.org/>

⁹<https://dl.acm.org/>

¹⁰<https://link.springer.com/>

¹¹<https://scholar.google.com/>

("Sentiment analysis" OR "text analysis" OR "opinion mining" OR "emotion AI")

AND

("software engineering" OR "development team" OR "software development" OR "software project")

We adjusted the search string according to the specific syntax of the data bases.

3.2.3 Definition of inclusion and exclusion criteria. During the review process, we eliminated studies and publications that cannot contribute to answering our research questions. In order to make this decision more objective, we defined inclusion and exclusion criteria as summarized in Table 1. We first applied the exclusion criteria to each of the found publications. In case that none of the exclusion criteria was true for the publication, we decided on the inclusion by considering the inclusion criteria. If the publication fits at least one inclusion criterion, it was included. If a relevant publication appeared more than once (e.g., as a conference paper and as an extended journal publication), we included the most recently published version.

Table 1: Inclusion and exclusion criteria

Inclusion
1. The publication presents an approach of the application of sentiment analysis in the context of SE.
2. The publication presents an approach to creating an sentiment analysis tool/algorithm in the context of SE.
3. The publication addresses the research questions of this SLR in its goals, hypothesis or applications.
4. This publication analyzes aspects related with this research.
Exclusion
1. The publication is not written in English.
2. The publication is not peer-reviewed.
3. The publication appears repeatedly. In this case, we only considered the latest version.
4. The publication is not accessible (respectively only accessible only by payment).
5. The publication has technical content without proven scientific relevance such as invitation papers, editorials, tutorials, keynotes, speeches, white papers, grey literature, dissertations, theses, technical reports, and books.
6. The publication is a document that is not a full paper or study such as presentations, web postings, web content, citations, brochures, pamphlets, newsletters, or extended abstracts.

3.2.4 Definition of quality assessments. In order to assess the quality of papers as objectively as possible, we defined quality assessments according to Kitchenham et al. [5, 35], which can be answered almost objectively. One example is whether a paper provides comprehensible conclusions or not. Depending on whether the criterion is fulfilled completely (=2), partially (=1) or not at all (=0), we assigned the scoring to the paper. If a criterion was not applicable, we

did not take it into account. If the average of all applicable quality attributes was less than 1, we considered the paper as having insufficient quality and removed it from further analyses. This happened five times.

3.2.5 Execution. An overview of the execution can be seen in Figure 1. First, we searched the five databases mentioned in subsection 3.2.1 using the search string and selected probably relevant papers based on their title from a total of 6763 publications. After removing duplicates and filtering the papers based on the titles, we applied the inclusion and exclusion criteria to the remaining 420 papers' abstracts. Ninety-nine papers remained of which we scanned the full text, resulting in 71 papers used as the startset for the forward and backward search. This snowball principle was performed according to Wohlin [75]. The backward search resulted in 24 new papers based on the title, and the forward search resulted in 8 papers. We again scanned the 32 new papers based on abstract and full text, leading to 14 more papers considered as relevant.

In total, we identified 85 papers as relevant: 71 in the initial search and 14 in one forward and backward search. The number of new papers after one snowball iteration was small. Therefore, we refrained from repeating the snowball principle after having examined the first papers and could not find any relevant new papers based on the title, except duplicates.

We ended our search for papers at the end of December 2020.

The quality metrics were applied to the 85 papers and five were excluded due to not passing the quality criterion. Some of them did not define research questions or discuss results, or the technologies used were not described. All of them had an average score less than 1 according to the quality assessments.

In this execution phase, we listed each relevant paper in an Excel spreadsheet [59]. We included relevant parts of the paper, and based on this data, we clustered the papers and created categories for the research questions.

3.2.6 Threats to Validity. The outcome of our literature review is biased by different factors. We cannot assume that we found all publications in the context of our research nor that our results are complete. In fact, since we found few papers that are not accessible, it is not unlikely that there are relevant papers that we could not include in our results.

In the following, we present threats to validity according to the different steps of the literature review.

Selection of databases: In order to find as many relevant papers as possible, a high number of used databases is an advantage. Some publications are listed in more than one database while others are not (construct validity). For our SLR, we used five scientific databases that are known to be relevant in the area of SE and have been used by several other SLRs in SE (e.g., [18, 36, 37, 64, 66, 77, 78]).

Definition of the search string: We composed the search string from the two areas *SE* and *sentiment analysis*. We used synonyms for *sentiment analysis*, as well as several words from the field of SE. Nevertheless, there may be other synonyms or related terms of which we are currently unaware (construct validity). Although the search string affects the outcome of the study, we are confident that the accuracy of the results found based on our search string suffices to answer our research questions and to achieve our goal of getting an overview on existing literature.

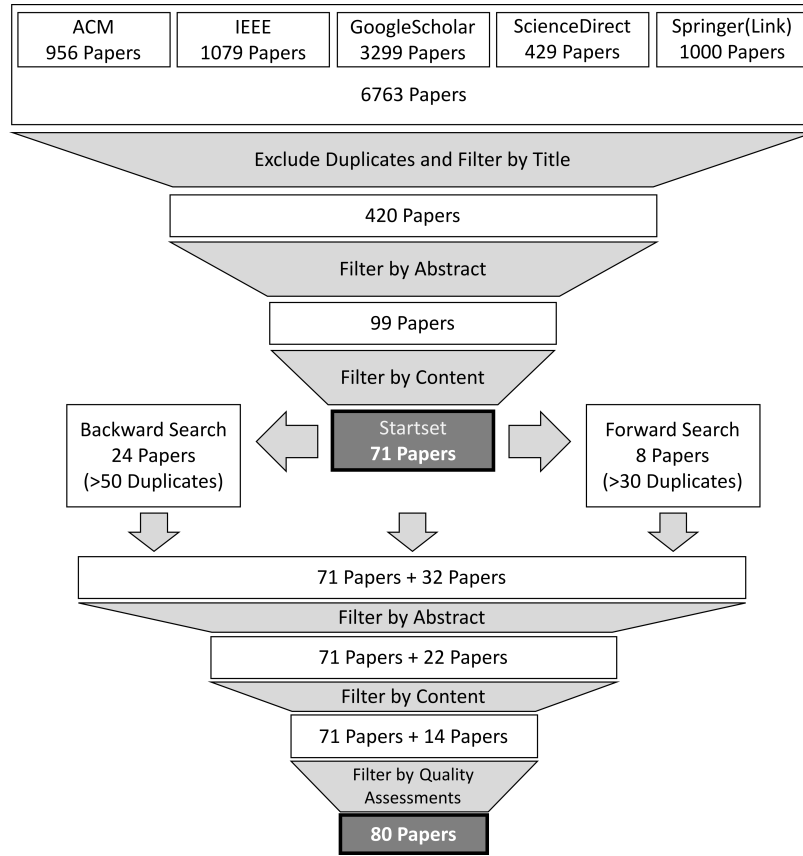


Figure 1: Search process and filtering steps

Definition of the inclusion and exclusion criteria: Certain publications can be included or excluded based on various characteristics. For example, we excluded papers that have not been peer-reviewed. In order to be as objective as possible and to reduce bias due to subjective decisions (internal validity), we formulated criteria for the inclusion and exclusion of a paper based on Kitchenham et al. [5, 35]. Some of the criteria like the accessibility of a paper are purely objective, while the evaluation of the content regarding the relevance to the study is still somewhat subjective. In case of doubts, we included the paper (and possibly excluded it later after having reviewed the abstract or content) and then decided how to proceed with the paper.

Definition of quality assessment: In addition to the inclusion and exclusion criteria, we set up quality assessments according to Kitchenham et al. [5, 35]. These assessments provide an objective framework to assess the quality of the paper and look at whether certain standards have been met, rather than arbitrarily including or excluding a paper. The lower limit for filtering out was chosen as 1, which means that all applicable quality criteria are on average at least partially fulfilled.

Execution: The study was largely conducted by one researcher, but the whole research process was reviewed by other researchers. Therefore, some researcher bias is present (internal validity), but we are confident that this influence only plays a minor role.

Results: Due to the design of the study, we cannot guarantee that we found all papers relevant to the research objective. In addition, a repeated application of the snowball principle would have found other possible relevant papers. However, we are confident that the papers we found are sufficient to answer the research questions.

4 RESULTS

Our literature review revealed 80 publications dealing with the topic of sentiment analysis in the context of SE. These publications are available online as raw data set [59].

4.1 Application Domain

During the classification phase based on the research question, we identified three application domains in which sentiment analysis was developed or applied: (1) open-source software (OSS) projects, (2) industry, and (3) academia. In case that the application domain was not explicitly named, we selected the domain based on the data sets used or the context of the usage of sentiment analysis. If, for example, GitHub was used, we aligned the publication to OSS domain. If data sets like app reviews were used, we chose the industrial domain.

Of the $n = 80$ papers, 59 belong to the domain of OSS, 17 to the domain of industry and 6 to the domain of academia. An example of application in an industrial context is a sentiment analysis on

developers chat communication data at Amazon MTurk [74]. University applications include sentiment analysis among students in software projects organized by universities [19, 22, 25].

Finding: Most of the papers on sentiment analysis in SE are based on open-source projects. Less than 1/3 of the papers either considers industrial projects or the academia.

4.2 Purpose

When investigating the motivation of the papers, we distinguish between three types of papers: (1) development, (2) comparison, and (3) application of sentiment analysis tools. These categories can be seen as primary motivation that can be refined further. The first type, *development*, consists of papers that aim to develop a sentiment analysis tool. The second type compares already existing tools with each other, while the third type focuses on their application. There are papers that developed a new tool and then compared it to existing tools. Since all these papers had the primary goal of developing a new tool, we assigned them to the type *development*.

Of the $n = 80$ papers, 16 belong to development, 12 to comparison and the remaining 52 to application. Details are summarized in Table 2. The papers of the type *development* have the main purpose of developing a procedure for sentiment analysis in order to examine certain data in the context of SE, such as developer communication (e.g. [8]), for sentiment. From $n = 12$ papers of the type *comparison*, 9 strive to find the best sentiment analysis tool among several tools. Three want to compare the allocation of sentiments from existing sentiment analysis tools with the allocation from humans. From $n = 52$ papers of the type *application*, 35 want to find correlations between sentiments and specific values, whereas 18 analyze social aspects of developers. Seven papers want to measure a specific value like subjective usability (e.g. [16]) or the marketability of an open-source app (e.g. [51]). Five papers use sentiment analysis tools, among other methods, to predict special values like the performance of a teacher based on feedback from students (e.g. [4, 33]).

Table 2: Motivations of the papers in detail

Type	Find best tool	Tools vs. human	Correlations	Social aspects	Values measurements	Values predictions
Comparison	9	3	0	0	0	0
Application	0	0	35	18	7	5

Finding: There are three types of papers, based on their main purpose: Development of sentiment analysis tools, comparison and application of them. Most of the papers belong to the application type, whereas 35% of the papers either belong to development or comparison.

4.3 Used data sources

Next, we analyzed which data is used for training or evaluation for sentiment analysis tools. We identified a total of 48 data sources. Most of them occur only one or two times, because the data used in the application domain are often unique (such as a specific data set of a student software project). Therefore, for the sake of clarity, we listed all data sets in Table 3 that occurred at least three times in total. Data sets that occurred less than three times are summarized as “other”.

Table 3: Overview of the used data

Type	JIRA	GitHub	Stack Overflow	App reviews	Twitter	Code reviews	Other
Development	6	3	9	0	2	1	3
Comparison	9	5	7	1	0	2	5
Application	6	12	4	7	2	0	38
Total	21	20	20	8	4	3	46

The papers from the development and comparison categories use the same data source for training and testing an algorithm. Only Chen et al. [10] use GitHub emojis for fine-tuning an existing neural network, but not for testing. It is notable that from the $n = 80$ papers, the data sources *JIRA*, *GitHub* and *Stack Overflow* are the most represented with 21, 20 and 20 uses respectively. App reviews were used 8 times, while all other data sources were used at most 4 times. In the application domain alone, from $n = 52$ papers, mostly unique (37 out of 38) data sources are used to apply a sentiment analysis tool to them. Other data sources include chat data from Amazon MTurk [74], Amazon product review [17], android bug reports [71], or support tickets from IBM [73].

Finding: The results show that there are 48 different data source. Six of them are used at least three times. JIRA, GitHub and Stack Overflow are used most frequently for training and testing sentiment analysis algorithms.

4.4 Approaches for developing or using sentiment analysis

In line with the results presented in subsection 4.2, we again distinguish between *development* and *application* of the tools when looking at the algorithms the sentiment analysis is based on. For example, when a paper is about the development of a sentiment analysis tool, we listed which machine learning approach this tool uses, such as SVM or Bayes. If an existing tool like SentiStrength [68, 69] is used, we listed this specific tool. Some papers did both developing and comparing their tool with other existing tools.

An overview of approaches being used during developing can be seen in Table 4. Because only existing tools were compared in the “comparison” category, we did not list the respective papers.

For the sake of clarity, we summed up all approaches that appeared less than three times in “other”.

Table 4: Overview of approaches used during developing sentiment analysis tools

Type	Bayes	SVM	Neural network	Random forest	Logistic regression	Decision tree	Lexicon/Heuristic	Gradient boosting	Other
Development	5	7	4	3	3	2	3	3	6
Application	6	4	3	2	1	1	0	0	1
Total	11	11	7	5	4	3	3	3	7

From the remaining $n = 68$ papers providing information on the used approaches, we found a total of 15 different machine learning approaches used for evaluations. With 11 times, SVM and different kind of Bayes classifiers (e.g. Naive Bayes) were used most frequently. However, also other established methods like random forest, logistic regression or AdaBoost were used, too. Three of the 15 machine learning methods were used only one time: Sequential minimum optimization [7], bootstrap aggregating [15] and pattern-based approach [41].

Some papers have also compared several approaches. From the $n = 68$ papers, 9 compared different machine learning approaches and chose the best performing one. In these comparisons, SVM won the comparison with three times the most. Gradient boosting is second and won two times. Bayes, random forest, logistic regression and neural network won only one time.

Table 5 gives an overview about the application of existing sentiment analysis tools. In the $n = 80$ papers, we found a total of 28 tools. For the sake of clarity, we summed up all tools that appeared less than three times in “other”.

Table 5: Overview of sentiment analysis tools applications. Tools marked with * are specifically designed for the SE domain)

Type	SentiStrength	NLTK	SentiStrength-SE*	Senti4SD*	CoreNLP	SentiCR*	Vader	Alchemy	WordNet	Other
Development	1	1	1	1	1	1	0	0	0	1
Comparison	8	6	5	5	4	5	1	3	1	12
Application	20	6	4	3	1	0	3	0	2	15
Total	29	13	10	9	6	6	4	3	3	28

The results show that three tools stand out, which were used at least 10 times. With 29 times, SentiStrength [68, 69] is the most

used sentiment analysis tool by far. The second most used tool is SentiStrenght-SE [28, 32], which is an adaptation of SentiStrength to the domain of SE. It was used 10 times. NLTK [46], which is a natural language toolkit and handles sentiment analysis, was used 13 times. All other sentiment analysis tools were used less than 10 times, often only 1-2 times total. These include, for example, the neural network BERT [14], the BERT-based model RoBERTa [45], the open source machine learning software WEKA [26] or the lexicon-based tool DEVA [30].

Finding: The results show that 28 different existing sentiment analysis tools were used in the $n = 80$ papers, with SentiStrength standing out. Concerning the different machine learning methods there are 15, which were used for evaluation in the $n = 80$ papers. SVM and Bayes stand out here. The authors often chose SVM because of its good performance.

4.5 Difficulties

Among all papers, some difficulties regarding the field of sentiment analysis in SE were mentioned frequently. Table 6 shows an overview of the results of the mentioned problems.

Table 6: Overview of mentioned problems regarding sentiment analysis in SE

Type	Adaption to the domain of SE	Sarcasm/Irony	Subjectivity of manual labeling	Small amount of data	Cross-platform performance
Development	9	3	0	4	1
Comparison	5	3	4	1	1
Application	12	5	6	1	2
Total	26	11	10	6	4

Of the $n = 80$ papers, 26 mentioned the problem of the lack or scarcity of adaptations of existing sentiment analysis tools to the domain of SE (e.g. [3, 8, 10, 15, 27]). However, other problems like sarcasm/irony handling (11) or the subjectivity of manual labeling of data (10) were also mentioned. In addition, there are also investigations of how different sentiment analysis tools perform, when they are trained in a cross-platform setting [54]. Novielli et al. [54] mention that the tools trained with one data set often performed poorly when they are tested with a different data set [54].

Finding: There are some difficulties regarding sentiment analysis in SE, which are mainly related to subjective data labels, too much customization for specific data sets and no/partial adaptation to the domain of SE.

5 DISCUSSION

5.1 Answer to research questions

RQ 1: What are the main application scenarios for sentiment analysis in the context of SE?

Our results show that there are three application domains: Open-source software domain, industry and academia. The $n = 80$ papers were most frequently classified in the OSS domain with 59, which is more than 2/3 of all papers. Examples for the OSS domain are studies using sentiment analysis on public data from platforms like GitHub, where OSS is developed (e.g. [55]). However, there are also case studies, where the sentiments of chats from developer teams are analyzed in the industry (e.g. [74]) or university (e.g. [19]).

RQ 2: For what purpose is sentiment analysis used in the investigated studies?

Based on their purpose, the studies on sentiment analysis in SE can be divided into three categories: Development, comparison and application. Regarding the sentiment analysis context, 52 from the $n = 80$ papers focus on applying a sentiment analysis tool. In most cases, the authors of the studies wanted to perform statistical analysis to find correlations between sentiments and a specific parameters (e.g. different times of a day in bug-introducing and bug-fixing commits [31]). Also, social aspects of developers were often studied. For example Whiting et al. [74] introduced a new technique that support online and remote teams in a way that the viability of the teams increase. In this context, they applied sentiment analysis on the developers' chats.

RQ 3: What data is used as a basis for sentiment analysis?

There are 48 different data sources, 6 of them are used at least three times. With 21, 20 and 20 usage respectively, JIRA, GitHub and Stack Overflow are used most frequently for training and testing sentiment analysis algorithms. Most of the data sets consist of commit/pull request comments, discussions, questions, or reviews.

When developing or comparing sentiment analysis tools, the authors of the papers often created or used data sets from platforms such as GitHub, JIRA or Stack Overflow. When using existing tools, these data sets also occurred, but most often the studies used individual data from specific development teams across all three application domains. This is also illustrated by the fact that we assigned over 3/4 of the data sets from the "other" category to papers of the application type. This category includes for example data sets that originate from the industrial sector (e.g. chat data from Amazon Mturk [74], or student chat data from software projects in universities (e.g. [22, 25])).

RQ 4: Which approaches are used when developing sentiment analysis tools?

We found 28 different existing sentiment analysis tools. SentiStrength [68, 69] stands out with 29 uses. This is probably due to the fact that the tool has been around for a while, has often been referenced as state-of-the-art, is domain-independent, and does not need to be trained as it uses a lexicon-based method.

Concerning the different machine learning methods, there are 15. SVM and Bayes stand out with 11 uses each. Neural Networks are third with 7 usages. The results indicate that SVM as well as

Naive Bayes are popular machine learning methods, which were most frequently tested in the context of a sentiment analysis tool. Senti4SD [8], which is the most commonly used tool that is not lexicon-based, implemented an SVM because it produced the best results. Studies that compared multiple machine learning methods ended up choosing SVM most often (3). Gradient boosting ended up in second place (2).

RQ 5: What are the difficulties of these approaches?

The authors often stated that existing, domain independent tools lead to poor results in the SE domain (e.g. [8, 27, 42]). This is because certain terms are used differently in the SE domain than in the non-technical context, resulting in different sentiments. In addition, the papers also described that labeling sentences with sentiment is often subjective, so people would already assign different labels to each other (e.g. [27, 42, 79]). Irony or sarcasm is also a problem that is mentioned (e.g. [28, 30, 52]), as a sentence can have a different sentiment when it is known to be meant ironically. Nevertheless, the listed problems should be put in a temporal context, as some of the problems have already been addressed and will be further addressed.

5.2 Interpretation

One possible explanation for the omnipresence of studies on OSS is that this data is available. This is supported by the fact that most of the data sets used for training or evaluation belong to the OSS domain. They are often from platforms like GitHub or Stack Overflow. This means that the public comments were mined and then manually labeled. But for the application of sentiment analysis in industry with the best possible performance, it would make sense to take data from the industry and train the tools with this data. Especially with the background that some papers have found that existing tools perform poorly in cross-platform settings.

One could see in the application category a tendency that there are still too few application scenarios in industry or that the tools do not yet seem interesting enough for the industrial context. Therefore, it might be necessary to investigate more intensively to what extent there is a demand for sentiment analysis in SE and to look into the reasons why it has not been used much so far in relation to OSS. One possible explanation for the scarce use in industry are legal and privacy issues. It is likely that developers have doubts if their data is analyzed according to the adequacy of the used language, and work councils often do not allow the analysis of existing data with respect to social aspects. Therefore, getting access to industry data sets is way more complicated than using data that is online available and can be used with less restrictions, if at all.

Regarding the usage of existing tools, SentiStrength [68, 69] is a well-established tool for sentiment analysis, because it has been used frequently in the application category (20) and has also constantly served as a comparison tool in the other two categories, development (1) and comparison (8). It is lexicon-based, which generally has the advantage of not having to be trained and it rarely has performance drops in different domains. The disadvantage, however, is that generic tools are not specialized for the respective domains like SE. Therefore, one of the first tools developed specifically for SE was SentiStrength-SE [28], which was adapted to the

SE domain. The advantage of the adaption to the SE domain and no need for training could explain why SentiStrength-SE [28] was used as the second most common. Other tools from the $n = 80$ papers are based on traditional machine learning approaches such as SVM or Bayes, which are the most common among all papers with 11 appearances. When trained in a specific domain, Senti4SD [8], for example, which uses SVM, achieves better accuracy than lexicon-based tools such as SentiStrength-SE [28], but then performs significantly worse on cross-platforms [54]. However, there are also tools based on artificial neural networks, like the BERT-based model RoBERTa [45], which sometimes outperforms Senti4SD [8] or other sentiment analysis tools [79].

Of the problems listed in Subsection 4.5, many were also attempted to be addressed by the papers, such as the adaption to the SE domain. The problem of irony or sarcasm, however, is a problem that is still not solved according to the results of our SLR. In addition, it is also mentioned that the existing tools mostly perform differently in cross-platform settings [2, 54, 70, 79]. This means that if, for example, a tool has been trained with a GitHub data set, it will perform well on that data, but may perform poorly on data from other platforms like Stack Overflow. This should be investigated further. Possible causes could be that the choice of words is communicated differently on the platforms, e.g. different levels of politeness. Or it may also be due to the labels of the different data sets, as some papers point out that people themselves often disagree about which polarity they assign to certain sentences and the assignment is ultimately also subjective [23, 27, 34, 42, 48, 49, 52, 54, 57, 60, 79].

5.3 Future research directions

To address the low application of sentiment analysis in the industrial context, it can be useful to create data sets that emerge directly from industry. Based on that data, sentiment analysis tools can be created, which will potentially perform better in the field of industry, because they are better adapted to practice. Nevertheless, there is a risk that the problems around subjective labeling and thus possibly also cross-platform performance will occur. To avoid this, it would therefore be meaningful to match the collected communication data with an additional regular sentiment survey of the developers. This way, it would be possible to compare the predictions of the trained tools with the collected manual sentiment data.

Possible solutions for the different cross-platform performance of the tools could be to examine the data sets and their labels for subjectivity, or to have them labeled according to the same emotion model like the Plutchik model [63] or the PANAS scale [72] instead of ad-hoc annotations. In addition, it might be meaningful to have it labeled by the same authors, leading to similar sets which are evaluated similarly and the bias is constantly contained in the form of a certain subjectivity, so that the tools perform more uniformly. Ad-hoc annotations may have the advantage of capturing perceived sentiments more accurately than a sentiment assignment based on emotion models. However, logically the subjectivity of an ad-hoc assignment is usually higher, so machine learning techniques then have their difficulties in achieving high performance with this subjective data. Another possibility would be to develop a tool that combines several well-performing tools to get a better result in a cross-platforming setting. Furthermore, an investigation of

the expressions and politeness levels of different platforms and subdomains would be useful. It should be investigated, for example, whether the developers on GitHub communicate in a different way than on Stack Overflow.

Logically, to address the problem of poor data, it makes sense to mine new data from the respective platforms like GitHub or Stack Overflow or even get it from the industry and label them.

The problem about irony/sarcasm handling is domain independent. Therefore, it might help to do a search of all new developments in natural language processing to see what new approaches are available to handle them.

Based on the results of our SLR, one can get good performance when using machine learning approaches like SVM or gradient boosting. However, our results do not contain enough data on sentiment analysis tools based on neural networks to draw conclusions about them. Nevertheless, according to the latest developments, this neural network approach delivers promising results [58, 79].

6 CONCLUSION

In development teams, there is a lot communication via various channels, so the social component of a developer plays a major role. Appropriate interaction with each other in these channels is therefore of great importance. For meetings, it has been proven that a negative mood quickly affects the entire team and everyone can be demotivated afterwards. Sentiment analysis tools help to counteract this in text-based communication. Since, to the best of our knowledge, no SLR has been conducted to provide an overview of various tools, their development, application and problems, we have conducted such a SLR.

We analyzed 6763 papers and found 80 relevant papers. We analyzed these papers according to the application area of sentiment analysis tools, the underlying data, procedures and purpose of application. One finding is that most papers only use sentiment analysis instead of developing it or comparing its performance with other tools.

We also identified several problems such as the handling of irony and sarcasm, the limited amount of data available to train and evaluate machine learning based tools, subjectivity in labeling data, and performance degradation in cross-platform settings. Some possible causes for the problems were mentioned as well as possible solutions. Future research should hence focus on objectively labeled data, especially on data from industry. The already existing data sets should be examined for different labels. In addition, combinations of existing tools can be considered in order to potentially achieve even better performance in a cross-platform setting.

REFERENCES

- [1] Mohamed Elhag M. Abo, Ram Gopal Raj, Atika Qazi, and Abubakar Zakari. 2019. Sentiment Analysis for Arabic in Social Media Network: A Systematic Mapping Study.
- [2] Md Ahasanuzzaman, Muhammad Asaduzzaman, Chanchal K. Roy, and Kevin A. Schneider. 2020. CAPS: a supervised technique for classifying Stack Overflow posts concerning API issues. *Empirical Software Engineering* 25, 2 (2020), 1493–1532. <https://doi.org/10.1007/s10664-019-09743-4>
- [3] Toufique Ahmed, Amiangshu Bosu, Anindya Iqbal, and Shahram Rahimi. 2017. SentiCR: A customized sentiment analysis tool for code review interactions. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 106–111. <https://doi.org/10.1109/ASE.2017.8115623>

- [4] Khin Zezawar Aung and Nyein Nyein Myo. 2017. Sentiment analysis of students' comment using lexicon based approach. In *16th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2017)*, Guobin Zhu (Ed.). IEEE, Piscataway, NJ, 149–154. <https://doi.org/10.1109/icis.2017.7959985>
- [5] Barbara Kitchenham and Stuart M. Charters. 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Vol. 2. https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering
- [6] Eshita Biswas, K. Vijay-Shanker, and Lori Pollock. 2019. Exploring Word Embedding Techniques to Improve Sentiment Analysis of Software Engineering Texts. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories*. IEEE, Piscataway, NJ. <https://doi.org/10.1109/msr.2019.00020>
- [7] Stefano Cagnoni, Lorenzo Cozzini, Gianfranco Lombardo, Monica Mordonini, Agostino Poggi, and Michele Tomaiuolo. 2020. Emotion-based analysis of programming languages on Stack Overflow. *ICT Express* 6, 3 (2020), 238–242. <https://doi.org/10.1016/j.icte.2020.07.002>
- [8] Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. 2018. Sentiment Polarity Detection for Software Development. *Empirical Software Engineering* 23, 3 (2018), 1352–1382. <https://doi.org/10.1007/s10664-017-9546-9>
- [9] Fabio Calefato, Filippo Lanubile, and Nicole Novielli. 2017. EmoTxt: A toolkit for emotion recognition from text. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. IEEE, Piscataway, NJ, 79–80. <https://doi.org/10.1109/ACIIW.2017.8272591>
- [10] Zhenpeng Chen, Yanbin Cao, Xuan Lu, Qiaozhu Mei, and Xuanzhe Liu. 2019. SEntiMoj: An Emoji-Powered Learning Approach for Sentiment Analysis in Software Engineering. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (Tallinn, Estonia) (ESEC/FSE 2019)*. Association for Computing Machinery, New York, NY, USA, 841–852. <https://doi.org/10.1145/3338906.3338977>
- [11] Munmun De Choudhury and Scott Counts. 2013. *Understanding Affect in the Workplace via Social Media*. Association for Computing Machinery, New York, NY, USA, 303–316. <https://doi.org/10.1145/2441776.2441812>
- [12] Mandar Deshpande and Vignesh Rao. 2017. Depression detection using emotion artificial intelligence. In *2017 International Conference on Intelligent Sustainable Systems (ICISS)*. 858–862. <https://doi.org/10.1109/ISSI.2017.8389299>
- [13] M.D. Devika, C. Sunitha, and Amal Ganesh. 2016. Sentiment Analysis: A Comparative Study on Different Approaches. *Procedia Computer Science* 87 (2016), 44–49. <https://doi.org/10.1016/j.procs.2016.05.124> Fourth International Conference on Recent Trends in Computer Science & Engineering (ICRTCSCE 2016).
- [14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
- [15] Jin Ding, Hailong Sun, Xu Wang, and Xudong Liu. 2018. Entity-level sentiment analysis of issue comments. In *2018 ACM/IEEE 3rd International Workshop on Emotion Awareness in Software Engineering*. IEEE, Piscataway, NJ. <https://doi.org/10.1145/3194932.3194935>
- [16] Alaa Mustafa El-Halees. 2014. Software Usability Evaluation Using Opinion Mining. *Journal of Software* 9, 2 (2014). <https://doi.org/10.4304/jsw.9.2.343-349>
- [17] Xing Fang and Justin Zhan. 2015. Sentiment analysis using product review data. *Journal of Big Data* 2, 1 (2015), 1–14. <https://doi.org/10.1186/s40537-015-0015-2>
- [18] Vahid Garousi, Kai Petersen, and Baris Ozkan. 2016. Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. *Information and Software Technology* 79 (2016), 106–127. <https://doi.org/10.1016/j.infsof.2016.07.006>
- [19] Andreas F. Gkontzis, Christoforos V. Karachristos, Chris T. Panagiotakopoulos, Elias C. Stavropoulos, and Vassilios S. Verykios. 2017. Sentiment Analysis to Track Emotion and Polarity in Student Fora. In *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, Vasileios Vlachos (Ed.). ACM, New York, NY. <https://doi.org/10.1145/3139367.3139389>
- [20] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. 2014. Happy software developers solve problems better: psychological measurements in empirical software engineering. *PeerJ* 2 (2014), e289. <https://doi.org/10.7717/peerj.289>
- [21] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. 2015. Do feelings matter? On the correlation of affects and the self-assessed productivity in software engineering. *Journal of Software: Evolution and Process* 27, 7 (2015), 467–487. <https://doi.org/10.1002/smr.1673>
- [22] Emitza Guzman. 2013. Visualizing emotions in software development projects. In *2013 First IEEE Working Conference on Software Visualization (VISOFT)*, Alexandru Telea (Ed.). IEEE, Piscataway, NJ. <https://doi.org/10.1109/vissoft.2013.6650529>
- [23] Emitza Guzman, Rana Alkadh, and Norbert Seyff. 2017. An exploratory study of Twitter messages about software applications. *Requirements Engineering* 22, 3 (2017), 387–412. <https://doi.org/10.1007/s00766-017-0274-x>
- [24] Emitza Guzman, David Azócar, and Yang Li. 2014. Sentiment analysis of commit comments in GitHub: an empirical study. In *11th Working Conference on Mining Software Repositories : proceedings : May 31 - June 1, 2014, Hyderabad, India*, Sung Kim, Martin Pinzger, and Premkumar Devanbu (Eds.). ACM, [Place of publication not identified], 352–355. <https://doi.org/10.1145/2597073.2597118>
- [25] Emitza Guzman and Bernd Bruegge. 2013. Towards Emotional Awareness in Software Development Teams. In *2013 9th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE) : proceedings : August 18-26, 2013, Saint Petersburg, Russia (Saint Petersburg, Russia) (ESEC/FSE 2013)*, Bertrand Meyer, Mira Mezini, and Luciano Baresi (Eds.). ACM, 671–674. <https://doi.org/10.1145/2491411.2494578>
- [26] G. Holmes, A. Donkin, and I.H. Witten. 1994. WEKA: a machine learning workbench. In *Proceedings of ANZIS '94 - Australian New Zealand Intelligent Information Systems Conference*. 357–361. <https://doi.org/10.1109/ANZIS.1994.396988>
- [27] Nasif Imtiaz, Justin Middleton, Peter Girouard, and Emerson Murphy-Hill. 2018. Sentiment and politeness analysis tools on developer discussions are unreliable, but so are people. In *2018 ACM/IEEE 3rd International Workshop on Emotion Awareness in Software Engineering*. IEEE, Piscataway, NJ. <https://doi.org/10.1145/3194932.3194938>
- [28] Md Rakibul Islam and Minhaz F. Zibran. 2017. Leveraging Automated Sentiment Analysis in Software Engineering. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories*. IEEE, Piscataway, NJ. <https://doi.org/10.1109/msr.2017.9>
- [29] Md Rakibul Islam and Minhaz F. Zibran. 2018. A comparison of software engineering domain specific sentiment analysis tools. In *25th IEEE International Conference on Software Analysis, Evolution and Reengineering*. IEEE, Piscataway, NJ. <https://doi.org/10.1109/saner.2018.8330245>
- [30] Md Rakibul Islam and Minhaz F. Zibran. 2018. DEVA: sensing emotions in the valence arousal space in software engineering text. In *Applied computing 2018*, Hisham M. Haddad, Roger L. Wainwright, and Richard Chbeir (Eds.). Association for Computing Machinery Inc. (ACM), New York, NY, 1536–1543. <https://doi.org/10.1145/3167132.3167296>
- [31] Md Rakibul Islam and Minhaz F. Zibran. 2018. *Sentiment Analysis of Software Bug Related Commit Messages*. ISCA. 3–8 pages. <https://www.researchgate.net/publication/3167132.3167296>
- [32] Md Rakibul Islam and Minhaz F. Zibran. 2018. SentiStrength-SE: Exploiting domain specificity for improved sentiment analysis in software engineering text. *Journal of Systems and Software* 145 (2018), 125–146. <https://doi.org/10.1016/j.jss.2018.08.030>
- [33] Phuripoj Kaewyong, Anupong Sukprasert, Naomie Salim, and Fatin Phang. 2015. The possibility of students' comments automatic interpreter using lexicon based sentiment analysis to teacher evaluation. https://www.researchgate.net/publication/285581082_THE_POSSIBILITY_OF_STUDENTS%27_COMMENTS_AUTOMATIC_INTERPRET_USING_LEXICON_BASED_SENTIMENT_ANALYSIS_TO_TEACHER_EVALUATION
- [34] Arvinder Kaur, Amrit Pal Singh, Gunet Singh Dhillion, and Divesh Bisht. 2018. Emotion Mining and Sentiment Analysis in Software Engineering Domain. In *Proceedings of the Second International Conference on Electronics, Communication and Aerospace Technology (ICECA 2018)*. IEEE, Piscataway, NJ, 1170–1173. <https://doi.org/10.1109/ICECA.2018.8474619>
- [35] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology* 51, 1 (2009), 7–15. <https://doi.org/10.1016/j.infsof.2008.09.009> Special Section - Most Cited Articles in 2002 and Regular Research Papers.
- [36] Jil Ann-Christin Klünder, Philipp Hohl, Nils Prenner, and Kurt Schneider. 2019. Transformation towards agile software product line engineering in large companies: A literature review. *Journal of Software: Evolution and Process* 31, 5 (2019), e2168. <https://doi.org/10.1002/smr.2168>
- [37] Mehmet Kosa, Murat Yilmaz, Rory O'Connor, and Paul Clarke. 2016. Software Engineering Education and Games: A Systematic Literature Review. *Journal of Universal Computer Science* 22 (12 2016), 1558–1574.
- [38] Robert E. Kraut and Lynn A. Streeter. 1995. Coordination in Software Development. *Commun. ACM* 38, 3 (March 1995), 69–81. <https://doi.org/10.1145/203330.203345>
- [39] Marco Kuhrmann, Paolo Tell, Jil Klünder, Regina Hebig, Sherlock A. Licorish, and Stephen G. MacDonell. 2018. HELENA Stage 2 Results. <https://doi.org/10.13140/RG.2.2.14807.52649>
- [40] Akshi Kumar and Arunima Jaiswal. 2020. Systematic literature review of sentiment analysis on Twitter using soft computing techniques. *Concurrency and Computation: Practice and Experience* 32, 1 (2020), e5107. <https://doi.org/10.1002/cpe.5107> e5107 CPE-18-1167.R1.
- [41] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, and Michele Lanza. 2019. Pattern-Based Mining of Opinions in Q&A Websites. In *2019 IEEE/ACM 41st International Conference on Software Engineering*. IEEE, Piscataway, NJ. <https://doi.org/10.1109/icse.2019.00066>
- [42] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, Michele Lanza, and Rocco Oliveto. 2018. Sentiment Analysis for Software Engineering: How Far Can We Go?. In *Proceedings of the 40th International Conference on Software Engineering (Gothenburg, Sweden) (ICSE '18)*. Association for Computing Machinery, New York, NY, USA, 94–104. <https://doi.org/10.1145/3180155.3180195>
- [43] Bing Liu. 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies* 5, 1 (2012), 1–167. <https://doi.org/10.2200/>

- S00416ED1V01Y201204HLT016
- [44] Bing Liu and Lei Zhang. 2012. *A Survey of Opinion Mining and Sentiment Analysis*. Springer US, Boston, MA, 415–463. https://doi.org/10.1007/978-1-4614-3223-4_13
 - [45] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.
 - [46] Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. <https://arxiv.org/pdf/cs/0205028>
 - [47] Jaafar Zubairu Maitama, Norisma Idris, and Abubakar Zakari. 2020. A Systematic Mapping Study of the Empirical Explicit Aspect Extractions in Sentiment Analysis. *IEEE Access* 8 (2020), 113878–113899. <https://doi.org/10.1109/ACCESS.2020.3003625>
 - [48] Alessandro Murgia, Marco Ortu, Parastou Tourani, Bram Adams, and Serge Demeyer. 2018. An exploratory qualitative and quantitative analysis of emotions in issue report comments of open source systems. *Empirical Software Engineering* 23, 1 (2018), 521–564. <https://doi.org/10.1007/s10664-017-9526-0>
 - [49] Alessandro Murgia, Parastou Tourani, Bram Adams, and Marco Ortu. 2014. Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In *11th Working Conference on Mining Software Repositories : proceedings : May 31 - June 1, 2014, Hyderabad, India*, Sung Kim, Martin Pinzger, and Premkumar Devanbu (Eds.). ACM, [Place of publication not identified]. <https://doi.org/10.1145/2597073.2597086>
 - [50] Sebastian C. Müller and Thomas Fritz. 2015. Stuck and Frustrated or in Flow and Happy: Sensing Developers' Emotions and Progress. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. 688–699. <https://doi.org/10.1109/ICSE.2015.334>
 - [51] Maleknaz Nayebi, Homayoon Farahi, and Guenther Ruhe. 2017. Which Version Should Be Released to App Store?. In *11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. IEEE, Piscataway, NJ, 324–333. <https://doi.org/10.1109/ESEM.2017.46>
 - [52] Nicole Novielli, Daniela Girardi, and Filippo Lanubile. 2018. A Benchmark Study on Sentiment Analysis for Software Engineering Research. In *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*. 364–375.
 - [53] Tuomas Niinimäki, Arttu Piri, and Casper Lassenius. 2009. Factors Affecting Audio and Text-Based Communication Media Choice in Global Software Development Projects. In *2009 Fourth IEEE International Conference on Global Software Engineering*. 153–162. <https://doi.org/10.1109/ICGSE.2009.23>
 - [54] Nicole Novielli, Fabio Calefato, Davide Dongiovanni, Daniela Girardi, and Filippo Lanubile. 2020. Can We Use SE-specific Sentiment Analysis Tools in a Cross-Platform Setting? *Proceedings of the 17th International Conference on Mining Software Repositories* (Jun 2020). <https://doi.org/10.1145/3379597.3387446>
 - [55] Nicole Novielli, Fabio Calefato, Davide Dongiovanni, Daniela Girardi, and Filippo Lanubile. 2020. A gold standard for polarity of emotions of software developers in GitHub. <https://doi.org/10.6084/m9.figshare.11604597>
 - [56] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. 2014. Towards Discovering the Role of Emotions in Stack Overflow. In *Proceedings of the 6th International Workshop on Social Software Engineering (Hong Kong, China) (SSE 2014)*. Association for Computing Machinery, New York, NY, USA, 33–36. <https://doi.org/10.1145/2661685.2661689>
 - [57] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. 2015. The Challenges of Sentiment Detection in the Social Programmer Ecosystem. In *Proceedings of the 7th International Workshop on Social Software Engineering - SSE 2015 (SSE 2015)*, Imed Hammouda and Alberto Sillitti (Eds.). ACM Press, New York, New York, USA, 33–40. <https://doi.org/10.1145/2804381.2804387>
 - [58] Nicole Novielli, Fabio Calefato, Filippo Lanubile, and Alexander Serebrenik. 2020. Assessment of SE-specific Sentiment Analysis Tools: An Extended Replication Study. <https://doi.org/10.1007/s10664-021-09960-w>
 - [59] Martin Obaidi and Jil Klünder. 2021. *Dataset: Systematic Literature Review on the Development and Application of Sentiment Analysis Tools in Software Engineering*. <https://doi.org/10.5281/zenodo.4726651>
 - [60] Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A. Visaggio, Gerardo Canfora, and Harald C. Gall. 2015. How can i improve my app? Classifying user reviews for software maintenance and evolution. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 281–290. <https://doi.org/10.1109/ICSM.2015.7332474>
 - [61] D. E. Perry, N. A. Staudenmayer, and L. G. Votta. 1994. People, organizations, and process improvement. *IEEE Software* 11, 4 (1994), 36–45. <https://doi.org/10.1109/52.300082>
 - [62] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. 2008. Systematic Mapping Studies in Software Engineering. BCS Learning & Development. <https://doi.org/10.14236/ewic/ease2008.8>
 - [63] Robert Plutchik. 1980. Chapter 1 - A GENERAL PSYCHOEVOLUTIONARY THEORY OF EMOTION. In *Theories of Emotion*, Robert Plutchik and Henry Kellerman (Eds.). Academic Press, 3–33. <https://doi.org/10.1016/B978-0-12-558701-3.50007-7>
 - [64] Nils Prenner, Carolin Unger-Windeler, and Kurt Schneider. 2020. How Are Hybrid Development Approaches Organized? A Systematic Literature Review. In *Proceedings of the International Conference on Software and System Processes (Seoul, Republic of Korea) (ICSSP '20)*. Association for Computing Machinery, New York, NY, USA, 145–154. <https://doi.org/10.1145/3379177.3388907>
 - [65] Kurt Schneider, Jil Klünder, Fabian Kortum, Lisa Handke, Julia Straube, and Simone Kauffeld. 2018. Positive affect through interactions in meetings: The role of proactive and supportive statements. *Journal of Systems and Software* 143 (2018), 59 – 70. <https://doi.org/10.1016/j.jss.2018.05.001>
 - [66] Stepan Shevtsov, Mihaly Berekmeri, Danny Weyns, and Martina Maggio. 2018. Control-Theoretical Software Adaptation: A Systematic Literature Review. *IEEE Transactions on Software Engineering* 44, 8 (2018), 784–810. <https://doi.org/10.1109/TSE.2017.2704579>
 - [67] Margaret-Anne Storey, Christoph Treude, Arie van Deursen, and Li-Te Cheng. 2010. The Impact of Social Media on Software Engineering Practices and Tools. In *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research (Santa Fe, New Mexico, USA) (FoSER '10)*. Association for Computing Machinery, New York, NY, USA, 359–364. <https://doi.org/10.1145/1882362.1882435>
 - [68] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2012. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology* 63, 1 (2012), 163–173. <https://doi.org/10.1002/asi.21662>
 - [69] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology* 61, 12 (2010), 2544–2558. <https://doi.org/10.1002/asi.21416>
 - [70] Parastou Tourani, Yujuan Jiang, and Bram Adams. 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* (Markham, Ontario, Canada) (CASCON '14). IBM Corp., USA, 34–44.
 - [71] Qasim Umer, Hui Liu, and Inam Illahi. 2020. CNN-Based Automatic Prioritization of Bug Reports. *IEEE Transactions on Reliability* (2020), 1–14. <https://doi.org/10.1109/TR.2019.2959624>
 - [72] David Watson, Lee Clark, and Auke Tellegen. 1988. Development and validation of brief measures of positive and negative affect: the PANAS scales. *Journal of personality and social psychology* 54, 6 (1988), 1063–1070. <https://doi.org/10.1037/0022-3514.54.6.1063>
 - [73] Colin Werner, Gabriel Tapuc, Lloyd Montgomery, Diksha Sharma, Sanja Dodos, and Daniela Damian. 2018. How Angry are Your Customers? Sentiment Analysis of Support Tickets that Escalate. In *2018 1st International Workshop on Affective Computing for Requirements Engineering*, Davide Fucci, Nicole Novielli, and Emitza Guzmán (Eds.). IEEE, Piscataway, NJ. <https://doi.org/10.1109/affectre.2018.00006>
 - [74] Mark E. Whiting, Irena Gao, Michelle Xing, N'godjigui Junior Diarrassouba, Tonya Nguyen, and Michael S. Bernstein. 2020. Parallel Worlds: Repeated Initializations of the Same Team to Improve Team Viability. 4, CSCW1, Article 067 (May 2020), 22 pages. <https://doi.org/10.1145/3392877>
 - [75] Claes Wohlin. 2014. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering* (London, England, United Kingdom) (EASE '14). Association for Computing Machinery, New York, NY, USA, Article 38, 10 pages. <https://doi.org/10.1145/2601248.2601268>
 - [76] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer, Berlin. <https://doi.org/10.1007/978-3-642-29044-2>
 - [77] Affan Yasin, Rubia Fatima, Lijie Wen, Wasif Afzal, Muhammad Azhar, and Richard Torkar. 2020. On Using Grey Literature and Google Scholar in Systematic Literature Reviews in Software Engineering. *IEEE Access* 8 (2020), 36226–36243. <https://doi.org/10.1109/ACCESS.2020.2971712>
 - [78] Kinza Zahra, Farooque Azam, Fauqia Ilyas, Huma Faisal, Nadia Ambreen, and Nida Gondal. 2017. Success Factors of Organizational Change in Software Process Improvement: A Systematic Literature Review. In *Proceedings of the 5th International Conference on Information and Education Technology* (Tokyo, Japan) (ICIET '17). Association for Computing Machinery, New York, NY, USA, 155–160. <https://doi.org/10.1145/3029387.3029392>
 - [79] Ting Zhang, Bowen Xu, Ferdian Thung, Stefanus Agus Haryono, David Lo, and Lingxiao Jiang. 2020. Sentiment Analysis for Software Engineering: How Far Can Pre-trained Transformer Models Go?. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 70–80. <https://doi.org/10.1109/ICSME46990.2020.00017>