Data Augmentation for Improving Emotion Recognition in Software Engineering Communication

Mia Mohammad Imran
Virginia Commonwealth University
Richmond, Virginia, USA
imranm3@vcu.edu

Yashasvi Jain
Drexel University
Philadelphia, Pennsylvania, USA
yj395@drexel.edu

Preetha Chatterjee
Drexel University
Philadelphia, Pennsylvania, USA
pc697@drexel.edu

Kostadin Damevski
Virginia Commonwealth University
Richmond, Virginia, USA
kdamevski@vcu.edu

ABSTRACT

Emotions (e.g., Joy, Anger) are prevalent in daily software engineering (SE) activities, and are known to be significant indicators of work productivity (e.g., bug fixing efficiency). Recent studies have shown that directly applying general purpose emotion classification tools to SE corpora is not effective. Even within the SE domain, tool performance degrades significantly when trained on one communication channel and evaluated on another (e.g, StackOverflow vs. GitHub comments). Retraining a tool with channel-specific data takes significant effort since manually annotating a large dataset of ground truth data is expensive.

In this paper, we address this data scarcity problem by automatically creating new training data using a data augmentation technique. Based on an analysis of the types of errors made by popular SE-specific emotion recognition tools, we specifically target our data augmentation strategy in order to improve the performance of emotion recognition. Our results show an average improvement of 9.3% in micro F1-Score for three existing emotion classification tools (ESEM-E, EMTk, SEntiMoji) when trained with our best augmentation strategy.

ACM Reference Format:
Mia Mohammad Imran, Yashasvi Jain, Preetha Chatterjee, and Kostadin Damevski. 2022. Data Augmentation for Improving Emotion Recognition in Software Engineering Communication. In *Proceedings of ACM Conference (ASE 2022).* ACM, Ann Arbor, Michigan, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Emotions can strongly impact activities that are collaborative in nature and require creativity and problem-solving skills, such as software development [1]. Recent research has shown that positive emotions (e.g., Joy) are associated with increased productivity and job satisfaction in software engineering teams [25, 27, 30, 46, 53]. On the other hand, negative emotions (e.g., Frustration) can cause developers to lose motivation and exhibit lower participation, ultimately leading to team attrition [29]. Negative emotions, which are known to impact cognitive processes, could also serve as an obstacle to learning a new programming language, code comprehension, etc. [59]. Thus, for quite a while, software engineering researchers have studied developer emotions and how they impact software development activities [60, 75, 79]. The goals of the research have been to empirically understand the causes and the impact of different emotional states on the productivity of an individual developer or a team [28, 29, 31, 50] and to design techniques that automatically detect developer emotion and provide recommendations to developers [11, 21, 26, 35, 39, 59, 70]. Success in achieving these goals is predicated on the ability to detect emotions with high accuracy.

While several approaches that target emotion detection in software developers' written text have been proposed, they have not been evaluated extensively due to the limited availability of manually annotated ground truth data [42]. Manual annotation of emotions in text is time consuming and requires additional effort to ensure annotator subjectivity is minimized [33, 42]. At the same time, there is ample evidence that emotion classifiers trained on general purpose data (i.e., not from software engineering) perform poorly in the software engineering domain, likely due to the specific vocabulary and characteristics of software engineering communication (e.g., occasional presence of short snippets of code) [8, 56]. In fact, emotion classification performance appears to be optimal when trained and evaluated on each specific software engineering channel separately, e.g., GitHub issue comments, StackOverflow comments [57].

In this paper, our aim is to understand the current limitations and improve emotion classification in software engineering written communication. More specifically, we address the following two research questions:

RQ1: *How effective are existing emotion classifiers in detecting emotions in GitHub comments? What types of emotions are most likely to be misclassified?*

To answer this RQ, we first create a new dataset for emotion classification based on GitHub issue and pull request discussions. We annotate the dataset based on the six emotion categories first introduced by Shaver[73], while also going beyond this categorization to a finer grained division using secondary and tertiary emotions. Using our dataset, we evaluate three of the most commonly used

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.
ASE 2022, October 2022, Ann Arbor, Michigan, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

tools for software engineering emotion classification (ESEM-E [51], EMTk [9], SEntiMoji [13]), showing that their accuracy is further reduced compared to the original datasets that the tools were built and evaluated for. We perform an error analysis focused on the instances that all of the three tools got wrong, in order to understand the limitations of the current generation of approaches. Our results indicate that a large number of the errors (the majority) are due to a simple inability of the tools to recognize clear lexical cues that are present in the text, and not due to more hard-to-discern causes, such as the emotion being implicitly expressed.

RQ2: *Can automatic data augmentation techniques be used to improve the effectiveness of existing emotion classifiers?*

To answer this RQ, we explore three different strategies for data augmentation. Each of the strategies significantly increases the training set size, generating 10x more training data than what we had at the outset. The strategies contrast between unconstrained augmentation, where we modify the original text in random places and without additional checks, and augmentation with a number of constraints that ensure it does not perturb the original emotion in the text. Our experimental results show that the best strategy focuses on preserving the emotional polarity of the text, which is positive for emotions like Joy and Love and negative for emotions like Anger. Focusing the augmentation directly on individual emotions is not as profitable, as we lack sufficiently large software engineering-specific lexicons that can be used to generate a large and diverse number of augmented instances.

Contributions: Improved emotion recognition taxonomy in developer written communication can impact empirical research as well as motivate new tools that improve emotion awareness in software engineering projects. To the best of our knowledge, we are the first to explore data augmentation strategies in order to deal with the scarcity of high quality annotated data and improve emotion recognition in software engineering-related text. A similar approach to ours could potentially also be leveraged to improve related tasks in software engineering, such as automatic generation of training data for sentiment analysis.

We publish the annotation instructions, annotated dataset, and source code to facilitate the replication of our study at: https://anonymous.4open.science/r/SE-Emotion-Study-0141/

2 BACKGROUND

Over the years, emotions have been conceptualized in different ways by software engineering researchers. We begin by providing background on measuring emotions and the existing annotated datasets. We then introduce data augmentation, a technique for dealing with data scarcity in machine learning by increasing the training set size.

2.1 Emotions in Software Artifacts

Leveraging research in psychology, over the years researchers have used several categorizations of emotions in written text. For instance, Ekman et al. [22] categorized emotions into six basic categories: Anger, Disgust, Fear, Joy, Sadness, and Surprise. On the other hand, Shaver et al. [73] identified six basic emotion categories: Love, Joy, Anger, Sadness, Surprise, and Fear. Shaver et al. expanded the basic set of emotions to secondary and tertiary levels in a tree-like structure. Shaver's emotional structure was refined in the work of Parrott et al. [62]. Plutchik [63] proposed a wheel-like structure of three layers of emotions with eight basic categories: Anger, Disgust, Fear, Joy, Sadness, Surprise, Trust, and Anticipation. Studies conducted by Cowen et al. identified 27 distinct categories based on 2185 short videos [16], 28 categories using facial expressions [17], and 24 using human vocalization [15]. Based on Cowen et al.'s studies, Demszky et al. devised 27 categories for text-based emotion recognition [19]. They also provided a mapping between these 27 categories and Ekman et al.'s six basic categories [22]. Although, it is widely accepted that emotions are comprised of basic categories that are combined to form more complex emotions (e.g, Frustration), there is no consensus on the complete list of categories that accommodate the wide-range of emotions observed in the software engineering domain [69].

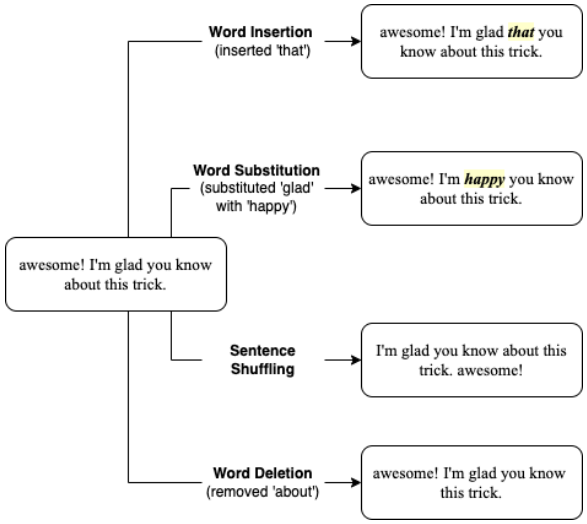


Figure 1: Example of data augmentation using four operators.

Available Datasets. To conduct studies of software developer emotions in various communication channels and software artifacts, researchers have created a few manually annotated datasets (i.e., gold sets) that leverage the categories described above. For instance, Ortu et al. [61] annotated a JIRA dataset that contained 5992 issue samples in three groups: group 1 contained 392 issue comments labeled with emotions Love, Joy, Surprise, Anger, Fear and Sadness; group 2 contained 1600 issue comments labeled with emotions Love, Joy, and Sadness; and group 3 contained 4000 issue sentences labeled with emotions Love, Joy, Anger, and Sadness. Novielli et al. [55] annotated a gold set from 4800 StackOverflow questions, answers, and comments. They labeled the sentences with Shaver et al [73]'s six basic categories. Venigalla et al. [82] analyzed 10996 commit messages related to software documentation update from 998 GitHub projected and mapped them into Plutchik's eight emotion categories [63].

2.2 Data Augmentation

Data augmentation (DA) [24, 41] is a technique for increasing training data diversity by targeted modification of the existing data.

Insufficient quantity of high-quality training data is a common problem in machine learning, especially with the emergence of more complex models, e.g., deep learning. The concept of DA originates in image processing, where researchers observed that, e.g., rotating an image by 90 degrees produces a new training instance for image classification tasks that increases the robustness of the model. Applying DA to written text has recently gained popularity despite the fact that this is usually a more difficult problem due to the complex relationship among written words. More specifically, DA has to maintain *label invariance*, that is, the augmented instance needs to have the same label as the original instance.

Data augmentation works by applying augmentation operators, one or more times to each training set instance. Popular and simple DA operators for text are word insertion, word substitution, sentence shuffling, word deletion, etc. Figure 1 shows an example data augmentation of an utterance from a GitHub issue comment using these operators. However, research shows that operators that target the specific classification task (i.e., emotion detection) are significantly more effective than generic ones [37]. Researchers also often use backtranslation (i.e., automatically translating to another language and back to English) and large language models (e.g., BERT) as DA operators, which are more likely to significantly change the original text, but with more risk towards perturbing the classification label.

3 METHODOLOGY

3.1 Data Selection

We selected four popular GitHub repositories, with each repository containing at least 50K stars. The repositories are: Flutter/flutter, Webpack/webpack, Microsoft/TypeScript, and Angular/angular. From each repository, we collected the last 10K comments (until 11 November, 2021) from pull requests and issues (5K pull request comments and 5K issue comments).

3.2 Preprocessing and Dataset Creation

We preprocessed each issue and pull request comment to replace the url, user mentions, and code with '<url>', '<username>', and '<code>' respectively. Consistent with previous research [8, 69], we did not remove stop words. We also did not include any additional preprocessing, since the emotion classification tools we use (ESEM-E [51], EMTk [9], SEntiMoji [13]) have their own preprocessing steps.

In a preliminary analysis, we observed that many instances in the GitHub comments are neutral, i.e., do not contain any emotion [52]. Our goal is to avoid creating a sparse dataset with mostly neutral instances, which would be inefficient and time-consuming to annotate. Novielli et al. also made a similar observation and performed selective sampling in creating their dataset of StackOverflow comments [55]. Hence, to avoid including too many neutral instances, we applied a software engineering-specific sentiment analysis tool [4] to label the preprocessed instances into 'positive', 'negative', and 'neutral'. For each of the four repositories, we randomly selected 250 pull request comments (125 'positive' comments and 125 'negative' comments), 250 issue comments (125 'positive' comments and 125 'negative' comments), to reach a total of 2000 instances.

3.3 Emotion Categories

As our primary emotion model, we use Shaver's framework [73] of emotion which has been commonly used in several software engineering studies [9, 51]. As shown in Table 1, Shaver's framework is a hierarchical (tree-structured) emotion representation model. There are three levels. At the top level, there are 6 basic emotion categories: Anger, Love, Fear, Joy, Sadness, and Surprise. For each of the basic emotions, there are secondary and tertiary level emotions, which refine the granularity of the previous level. For example, Optimism and Hope are the secondary and tertiary level emotions for Joy, respectively.

We observed that some commonly expressed emotions in developer communications, such as Approval, Disapproval, Confusion, Curiosity, etc., are missing from Shaver's framework, which was not designed for emotions expressed in text. For example, the following GitHub comment can be categorized with the emotion Curiosity, *"I'm curious about this - can you give more context on what exactly goes wrong? Perhaps if that causes bugs this should be prohibited instead?"*, but it does not clearly fit into any of Shaver's existing categories. Therefore, to accommodate a wider range of emotions observed in our dataset, we use the recent text-based emotion classification framework by Demszky et al., known as GoEmotions [19]. GoEmotions uses 27 emotions to annotate Reddit comments, which are mapped to Ekman's [22] 6 basic emotion categories. Note that, 5 of Shaver's basic emotions (Anger, Fear, Joy, Sadness, and Surprise) are the same as Ekman's basic categories. Ekman's basic category Disgust is a secondary emotion of Anger in Shaver's categories, and Shaver's basic category Love is a subcategory of Joy in Ekman's basic categories.

To enhance Shaver's categories, we include selected emotions from GoEmotions [19] (as shown in Table 1). We adopted GoEmotions's definitions and mapping only when an emotion is missing and does not conflict with Shaver's framework. The six additional secondary emotion categories that we adopted from GoEmotions [19] are highlighted in blue in Table 1. Note that all the tertiary level emotions we used come directly from Shaver's framework [73]; we did not add additional tertiary level emotions.

3.4 Data Annotation

Pull request and issue *comments* in GitHub usually consist of multiple sentences. While sometimes each sentence may express different emotions, more often, the comment as a whole conveys an unique emotion. Therefore, in this study, we consider comment-level granularity for data annotation.

Two human judges (authors of this paper) were provided a set of GitHub comments with annotation instructions as follows:
"*You will use the coding schema reported in Table 1. For each row in the spreadsheet, please indicate what emotion it conveys (if any) among the basic emotions (first column in the table), which are: Anger, Love, Fear, Joy, Sadness, Surprise. Multiple emotion labels from the basic emotions are allowed but you should try to avoid them if possible. You can use the second and third levels in the schema as a reference for choosing the primary emotion, but the annotation should be only for the primary emotions. Please mention the second and third-level emotions whenever they are prevalent, and provide a rationale for each annotation. Make sure you consider the emotion(s) of the entire*

Table 1: Shaver's [73] tree-structured emotion categories (additional categories from GoEmotions [19] shown in blue)

Basic Emotion	Secondary Emotion	Tertiary Emotion
Anger	Irritation	Annoyance, Agitation, Grumpiness, Aggravation, Grouchiness
	Exasperation	Frustration
	Rage	Anger, Fury, Hate, Dislike, Resentment, Outrage, Wrath, Hostility, Bitterness, Ferocity, Loathing, Scorn, Spite, Vengefulness
	Envy	Jealousy
	Disgust	Revulsion, Contempt, Loathing
	Torment	-
	Disapproval	-
Love	Affection	Liking, Caring, Compassion, Fondness, Affection, Love, Attraction, Tenderness, Sentimentality, Adoration
	Lust	Desire, Passion, Infatuation
	Longing	-
Fear	Horror	Alarm, Fright, Panic, Terror, Fear, Hysteria, Shock, Mortification
	Nervousness	Anxiety, Distress, Worry, Uneasiness, Tenseness, Apprehension, Dread
Joy	Cheerfulness	Happiness, Amusement, Satisfaction, Bliss, Gaiety, Glee, Jolliness, Joviality, Joy, Delight, Enjoyment, Gladness, Jubilation, Elation, Ecstasy, Euphoria
	Zest	Enthusiasm, Excitement, Thrill, Zeal, Exhilaration
	Contentment	Pleasure
	Optimism	Eagerness, Hope
	Pride	Triumph
	Enthrallment	Enthrallment, Rapture
	Relief	-
	Approval	-
	Admiration	-
Sadness	Suffering	Hurt, Anguish, Agony
	Sadness	Depression, Sorrow, Despair, Gloom, Hopelessness, Glumness, Unhappiness, Grief, Woe, Misery, Melancholy
	Disappoint	Displeasure, Dismay
	Shame	Guilt, Regret, Remorse
	Neglect	Embarrassment, Insecurity, Insult, Rejection, Alienation, Isolation, Loneliness, Homesickness, Defeat, Dejection, Humiliation
	Sympathy	Pity
Surprise	Surprise	Amazement, Astonishment
	Confusion	-
	Curiosity	-
	Realization	-

comment and not of individual sentences.". The annotation instructions contained more details of the schema including definitions and examples for the basic six emotions.

The judges initially annotated a shared set of 400 comments. The sample size of 400 is sufficient to compute the annotator agreement measure with high confidence [6]. The two annotators manually labeled the comments and measured Cohen's Kappa inter-rater agreement for the six basic emotions. For each of the emotions, they found agreement greater than 0.8, which is considered to be sufficient (> 0.6) [77]. The annotators further discussed their annotations until all disagreements were resolved. Afterwards, the annotators separately annotated 800 instances each to reach a total of 2000 utterances. Figure 2 shows the distribution of basic emotion categories per project in the final annotated set. In total, our dataset consists of 310 instances of Anger, 220 instances of Love, 198 instances of Fear, 422 instances of Joy, 274 instances of Sadness, and 328 instances of Surprise.

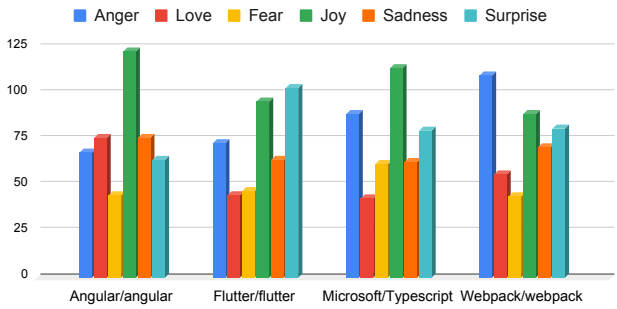


Figure 2: Frequency of emotions per project.

3.5 Studied Emotion Classification Tools

We investigate three existing software engineering-specific emotion classification tools, which we describe as follows:

ESEM-E [51]: Murgia et al. proposed a emotion classification tool, which was later referred to as ESEM-E. ESEM-E used Parrott's emotion categories as classification targets [62], which are also featured in Shaver et al.'s model [73]. While the source code of ESEM-E is not publicly available, we carefully read the descriptions detailed in the paper and implemented it. ESEM-E uses unigram and bigram features and machine learning (ML) models such as SVM, Random Forest, KNN, etc. As recommended by the authors, we use the SVM model.

EMTk [9]: Calefato et al. proposed EMTk (also known as EmoTxt), which was designed to identify developer emotions from textual communication channels. EMTk identifies six primary emotions according to Shaver's framework [73]. The implemented tool is publicly available on GitHub. EMTk provides two types of data sampling, 'NoDownSampling' and 'DownSampling'; 'DownSampling' randomly samples the majority class to balance the amount of instances between the majority and minority class, while 'NoDownSampling' does not change the training data. We use 'NoDownSampling' to ensure that all of the three tools use the same training data set.

SEntiMoji [13]: Chen et al. proposed SEntiMoji, a transfer learning approach for emotion detection in software engineering (SE) text based on emojis. They concluded that SEntiMoji can significantly outperform existing emotion detection methods (e.g., DEVA [35], EMTk [9], MarValous [34], ESEM-E [51]) in software engineering. The SEntiMoji source code is publicly available in GitHub. SEntiMoji is developed based on DeepMoji [23] which is

an existing deep learning based emoji representation model. The SEntiMoji model can identify various different emotion categorization schemas including Shaver's framework [73].

3.6 Metrics

We choose popular metrics used to evaluate a classification task: *Precision*, *Recall* and F1-score, which aggregates the prior two. In places where we present combined results across all emotions, we use the micro-averaged variants each of the metrics, as they are responsive to the frequency of occurrence of each constituent emotion.

- Precision: Precision is the ratio of true positive observations to the total predicted positive observations.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

- Recall: Recall is the ratio of true positive observations to all observations in the class yes.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

- F1-score: F1-score is the weighted average of Precision and Recall.

$$F1 - score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

3.7 Experiment Design

Using random stratified sampling [5] for each basic emotion, we divide our annotated dataset into train (80%) and test (20%) set. The test set contains a total of 400 data points including 68 instances of Anger, 44 instances of Love, 40 instances of Fear, 84 instances of Joy, 55 instances of Sadness and 65 instances of Surprise.

4 RQ1: EXISTING EMOTION CLASSIFIERS

RQ1: *How effective are existing emotion classifiers in detecting emotions in GitHub comments? What types of emotions are most likely to be misclassified?*

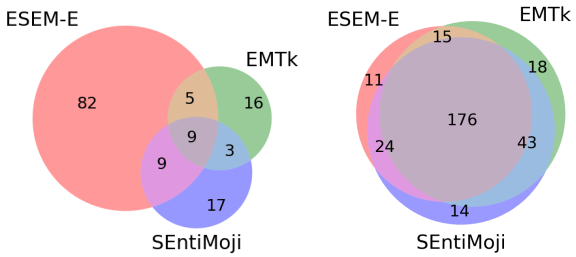
4.1 Classification Results

To answer this RQ, we evaluate three well-known tools for software engineering emotion classification (ESEM-E [51], EMTk[9], SEntiMoji[13]), on our dataset of GitHub issue and pull request discussions (described in Section 3). The per-emotion performance of the emotion detection tools is summarized in Table 2. The overall trend among all tools is for precision to be significantly higher than recall. In other words, the tools are acting conservatively, choosing to predict more utterances as negative (lacking a certain emotion) than positive, which leads to lower recall. Based on the aggregate measure, F1-score, ESEM-E performed best for Love, Joy and Sadness, EMTk performed best for Fear and Surprise, and SEntiMoji performed best for Anger.

Results across all emotions are summarized in the bottom part of Table 2. Here, on the micro-averaged F1-score metric, ESEM-E improves over the next best tool EMTk by 0.018 (4.3%). On micro-averaged precision, EMTk improves over SEntiMoji by 0.036 (5.0%), while on micro-averaged recall, ESEM-E improves the next best tool EMTk by 0.073 (25.0%). While ESEM-E performs best by far on recall, its performance on precision is worse than EMTk by 0.206 (37.3%) and SEntiMoji by 0.17 (30.7%). Overall, across three tools, the average micro F1-score is 0.421.

Table 2: Comparison of emotion detection tools on GitHub data.

Emotion	Model	Precision	Recall	F1-score
Anger	ESEM-E	0.405	0.250	0.309
	EMTk	0.571	0.118	0.200
	SEntiMoji	0.600	0.265	**0.367**
Love	ESEM-E	0.651	0.636	**0.644**
	EMTk	0.786	0.500	0.611
	SEntiMoji	0.733	0.500	0.595
Fear	ESEM-E	0.533	0.200	0.291
	EMTk	1.00	0.200	**0.333**
	SEntiMoji	0.714	0.125	0.213
Joy	ESEM-E	0.458	0.321	**0.378**
	EMTk	0.640	0.190	0.294
	SEntiMoji	0.609	0.167	0.262
Sadness	ESEM-E	0.759	0.400	**0.524**
	EMTk	0.778	0.382	0.512
	SEntiMoji	0.857	0.327	0.474
Surprise	ESEM-E	0.596	0.431	0.500
	EMTk	0.823	0.446	**0.580**
	SEntiMoji	0.846	0.338	0.484
Overall	ESEM-E	0.553	0.365	**0.440**
	EMTk	0.759	0.292	0.422
	SEntiMoji	0.723	0.278	0.402
	Average	0.678	0.312	0.421



(a) False positive. **(b) False negative.**

Figure 3: Distribution of FPs and FNs across different tools.

To examine whether the tools tend to struggle on the same instances or if they have complementary strengths, we plot Venn diagrams of the false positive and false negative instances in Figure 3. While the false positive instances seem to be broadly spread across different tools, the vast majority of false negatives instances are shared among the three tools. In total, 176/301 (58%) false negative instances across all emotions were misclassified by all three tools.

4.2 Error Analysis of FNs

Because of the unusually high agreement between the tools on the false negative (FN) instances, we focus our error analysis there, i.e., on the 176 FN instances.

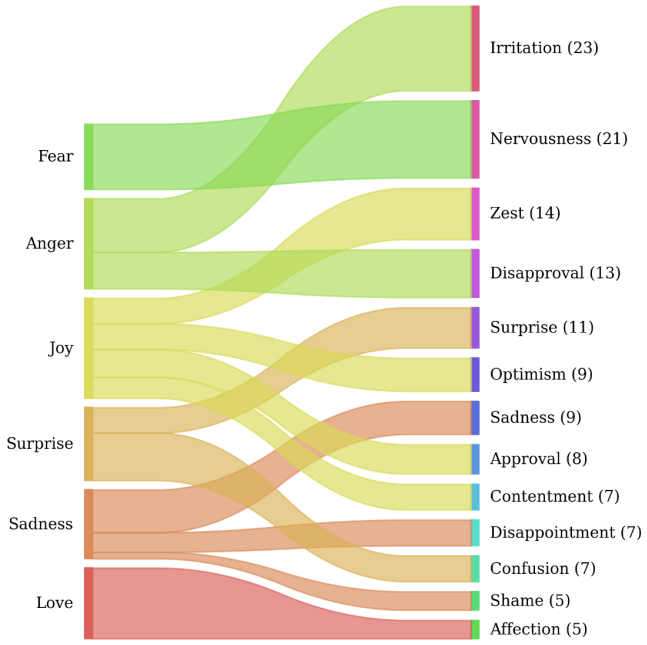


Figure 4: FNs mapped to their secondary emotions ($n >= 5$).

First, we examine the secondary emotions (as listed in Table 1) that are present in the FNs with the goal of understanding if specific emotions are particularly difficult to classify. We create a visualization in Figure 4 to understand the distribution of FN instances, i.e., to create the mapping of secondary emotion categories (right side of image) to the six basic level emotions (left side of image) in the FN instances. In this visualization, we only consider secondary emotion categories which has at least 5 FNs. The width of the ribbons of top level emotions represent their proportions in the dataset, while the width of the ribbons on the right side represent their proportion of FNs. We observe that some secondary emotions like Irritation, Nervousness and Zest have a significant number of FNs and represent the majority of FNs for basic emotions like Anger, Fear and Joy. For instance, Irritation expressed via comments like *"oh my god, explanation of official documents waste eight hours of me. Why isn't there a case to explain this"*, was misclassified (as FN) 23 out of the 34 times (67.6%) it appeared in the test set. Nervousness, e.g., *"I guess my concern is that it sets a precedent where somebody could see it and think that it would be fine to use in core"*, was also difficult to recognize as it was misclassified 21 out of 32 times (65.6%), while Zest was also mistaken often, with a misclassification frequency of 14/18 (77.8%). Relative to these hard to recognize emotions, Affection, which is part of the Love basic emotion, was a FN only 5 out of 35 times (14.3%).

Second, to understand the specific difficulties that the tools encountered, we performed a manual qualitative analysis of the 176 FNs. To perform this analysis, we use the error categories defined by Novielli et al. [58] to understand sentiment classification errors in software engineering text.

For each of the FN instances in our dataset, one of authors of this paper performed the initial error mapping, while another author reviewed it and indicated disagreements that were resolved via a discussion. In Table 3, we report the distribution of error categories assigned to our FN instances. During the analysis, we observed that multiple error categories can be assigned to some of the FN instances in our dataset. Hence, we chose more than one (i.e., two) error categories for 16 FN instances, while the rest 160 (176 - 16) instances were assigned one error category each.

The most frequent error category found in the FNs is *General error*, which indicates an inability to recognize lexical cues that occur in the text. For instance, in the comment *"that's awesome, I've been needing this for a while"*, annotated as Joy, the tools likely missed clear lexical cues (e.g., the word "awesome"). Similarly for *"oh my god, explanation of official documents waste eight hours of me, Why isn't there a case to explain this"*, annotated as Anger, the tools likely missed the idiomatic expression "waste N hours of me". In other cases, the classifiers miss due to misspellings or broken syntax, as in *"It's anoying me specifically when I want to set it as default value in constructors"*, which is annotated as Anger. *General error* is also the most prevalent in 10 of the 13 secondary categories that are shown in Figure 4.

In 61 cases, the tools failed because of the presence of *Implicit sentiment polarity* in texts. Often, humans use common knowledge to recognize emotions that the tools miss. Consider the following example *"Specifically, I'm less confident in the second commit than the first. AFAICT, it could only return true if a recursive call to itself returned true and all of the recursive base cases returned false."*. This was annotated as Fear (annotators perceived it as an expression of Worry – a 3rd level emotion that maps to Fear, see Table 1), but the emotion is not present explicitly. Sometimes, annotators inferred an emotion based on external knowledge. For instance, *"In that case I would advise you to please file a separate issue with the exact steps and logs to reproduce the issue. Because this issue is about existing apps. Thanks"*, was annotated as Anger since the speaker is expressing a violation of the community rules (Irritation as the secondary emotion). *Implicit sentiment polarity* is the most prevalent in 4 our of the 13 secondary emotions in Figure 4 (one category was a tie with *General error*). As reported by Novielli et al. [56], hostile attitude is often implicit and indirect, which we observe in the error for Anger's secondary emotions like Exasperation and Irritation. Demszky et al. [19] noted that the Nervousness emotion is likely expressed implicitly, which we also observe in our data; the top error category of comments marked with Nervousness is *Implicit sentiment polarity*.

In 15 cases, we notice that the tools were not able to correctly classify utterances due to *Pragmatics*. This type of error occurs when the annotators consider the context of the comment. In the comment *"hmm, even after a push I still see this test on github, but not locally"*, the author seems to have encountered something unexpected, which the annotators marked as Surprise.

Sometimes, the use of *Figurative language*, such as humor, irony, sarcasm, or metaphors, causes difficulties for the classifiers to identify emotions correctly. Often this type of utterances use neutral words to express an emotion. For example, *"Well, if you tried it you'd know"*. In other cases, the lexical presence of *Politeness*, such as "thank you", "please", etc., may cause misclassification. For instance, consider the following example, *"Hi, thanks for your contribution, but we can't review this because you didn't follow the contributing*

Table 3: Distribution of the error categories (as defined by Novielli et al. [58]) in the FN instances.

Error category	Count
General error	77
Implicit sentiment polarity	61
Pragmatics	15
Figurative language	15
Politeness	10
Polar facts	8
Subjectivity in annotation	6

instructions [...]", which is marked as Anger due to the violation of community rules (secondary emotion - Irritation). In other cases, the utterances involve *Polar facts*, that is the utterance invokes an emotion for most people, i.e., the annotators consider the reported situation to invoke an emotion. For instance, in *"I blame the autoformatter."*, the annotators marked this comment as Anger as they considered the author was irritated for facing same problem (second level emotion - Irritation). Overall, we observe that *Figurative language*, *Politeness* and *Polar facts* usually occur in negative emotions (i.e., Anger, Sadness, Fear). The annotation in emotion and sentiment is a subjective task [71] as the perception of emotions varies depending on personality trait and personal relevant dispositions. We observe this in 3.1% cases in our error distribution. For instance, *"Do you understand that it is impossible in some cases or can lead to increase size of bundle?"* was considered as Regret (3rd level Sadness) by one annotator, however, the other annotator considered it "Neutral".

Takeaway: Towards improving or designing new emotion classification tools, some types of errors could be more difficult to address than others. We hypothesize that tools should be able to improve on the most prevalent category of *General error* the most, as the lexical cues can be introduced via better training data, which could then be recognized by the tools. Towards that goal, we next investigate if *Data Augmentation* can be an effective strategy to automatically build better training data.

5 RQ2: DATA AUGMENTATION

RQ2: *Can automatic data augmentation techniques be used to improve the effectiveness of existing emotion classifiers?*

5.1 Augmentation Strategies

We explore three different data augmentation strategies that target emotion classification in software engineering. For each strategy, we use augmentation operators that transform each instance from the training set into a number of augmented instances, each introducing a slightly different vocabulary or idioms into the training set. In fact, we augment by applying a randomly chosen set of our augmentation operators, one after the other in a "stacked" fashion. In some of the augmentation operators, we rely on recently-introduced generative techniques that are capable of introducing realistic word spans [18, 38].

Via the different augmentation strategies we propose, we explore unconstrained vs. constrained choices of augmented data. For instance, we examine software-specific vs. generic choices of words to augment with, and how to ensure the original emotions are preserved (or enhanced) by the augmentation. Specifically, we introduce the following three data augmentation strategies: *Unconstrained*, *Lexicon-based*, and *Polarity-based*.

Unconstrained Strategy. The unconstrained strategy uses augmentation operators that have been previously shown to be effective in NLP and applies them at a randomly chosen location in the text. Inspired by Kumar et al.'s [38] work where they found that a BART-based [40] generative model outperformed other strategies, we use BART to create generative augmentation operations such as Word Insertion and Word Substitution. The Unconstrained Strategy uses the following four operators:

- *Word Insertion using BART:* We insert a word at any position in the original utterance.
- *Word Substitution using BART:* We substitute a word at any position.
- *Word Deletion:* We randomly delete a word at any position.
- *Sentence Shuffling:* When an utterance has more than one sentence, we randomly shuffle the sentences.

Lexicon-based Strategy. We observe that sometimes the Unconstrained Strategy produces utterances that may not preserve the original emotion. Note that one of the primary requirements of data augmentation is label invariance, i.e., for the original label to be preserved through the transformation. To deal with this problem, we leverage a software engineering-specific emotion lexicon [45] in order to validate the augmented words generated through the Unconstrained Strategy. Specifically, for each augmented utterance produced by the Unconstrained Strategy, we check if the augmented words exhibit an emotion and if the word's emotion does not match the original emotion. In that case, we replace the word with a software engineering emotion-specific word that preserves the original emotion of the instance. If an utterance is annotated as Joy, and an augmented word exhibits a different emotion (and not Joy), we replace the word with a word from the Joy category of a software engineering-specific lexicon. For example, the Unconstrained Strategy augments the following utterance, which is annotated as Love, *"This looks good, thanks for clarifying the docs."* to *"This looks worse, thanks for reviewing the docs."*. Here the introduction of the word *"worse"* changes the emotion of the original. However, if *"worse"* is replaced with a Love-specific word, i.e., *"wonderful"*, the text becomes *"This looks wonderful, thanks for reviewing the docs."*, which preserves the original label.

As a lexicon, we use NRC's [48] emotion lexicon combined with the software engineering-specific emotional lexicon from Mäntylä et al.'s work [45], which contains a total of 428 words. Since Mäntylä et al.'s lexicon is not annotated with Shaver's basic emotion categories, we use NRC's emotion lexicon to map each word from Mäntylä et al.'s lexicon to Shaver's basic categories. For example, Mäntylä et al.'s lexicon contains the word *afraid*, which is also available in the NRC emotion lexicon. Since each word in the NRC lexicon is annotated with a specific category (e.g., the word *afraid* is annotated under the emotion category Fear), we map these words to Mäntylä et al.'s lexicon to get a lexicon that is software engineering-specific and also has associated emotion categories.

Note that, as the NRC emotion lexicon uses Plutchik's [63] emotion categories which has 8 basic emotions, we make two adjustments. First, their basic emotion Disgust is a subcategory in Shaver's basic emotion Anger. Therefore, we combine NRC's Disgust module with Anger module and use it in our Anger lexicon. Second, Plutchik's categories do not contain Shaver's basic category Love, therefore we use NRC's positive module instead as Love; the positive module contains words with positive polarity.

Polarity-based Strategy. While the Lexicon-based Strategy removes some of the noise that is introduced by the Unconstrained Strategy, we believe the process can be streamlined and the augmentation quality further improved. For instance, a significant problem with the Lexicon-based Strategy is that it uses a lexicon with a very limited number of words. To overcome this constraint, instead of specific emotions, we focus on increasing the polarity words in the augmented instances. Inspired by GoEmotions' [19] grouping of emotions with sentiment polarity, we formulate three rules that augmented instances have to follow: 1) preserve (or increase) positive polarity words when the annotated utterance is Love and Joy, 2) preserve (or increase) negative polarity words when the annotated utterance is Anger, Fear and Sadness, and 3) preserve the original utterance polarity when the annotated utterance is Surprise.

To identify words that exhibit positive polarity, negative polarity or no polarity, we use SentiWordNet 3.0 [3]. While ensuring that each valid instance follows the above criteria, we generate new augmented instances using the same operators as for the Unconstrained Dataset. The only modification is that for *Word Deletion*, we only randomly delete a word if it does not exhibit sentiment polarity.

5.2 Augmentation Process

For all three of the above data augmentation strategies, for each instance in our training set, we generate 10 augmented instances, which is considered a reasonable augmentation ratio in the literature [66]. For each generated instance, if *Sentence Shuffling* is used, we only apply it once. We apply *n* augmentation operations to each instance, where *n = max(2, 20% of the length (i.e., number of words in the instance))* [87]. We use *nlpaug* [44] for the generative operations and use *bart-base* [40] as our BART model's weights. To further ensure that the augmented instance does not change the meaning of the original instance, we added an additional quality check where we ensure that the cosine similarity of BERTOverflow [78] vectors computed from the augmented and original instance are (>= 0.9) apart [65]. BERTOverflow is software engineering-specific version of BERT [20], pre-trained on the StackOverflow data dump. We load BERTOverflow using the *huggingface* library [84].

5.3 Augmentation Results and Discussion

Overall, across all three tools, all of the augmentation strategies improved performance over the original results (Table 4). The average micro F1-score improvement with the Unconstrained Strategy is 4.8% (0.441), with the Lexicon Strategy is 7.8% (0.455), and with the Polarity Strategy is 9.3% (0.461). Considering the three tools separately, we observe improved F1-score across the board, with EMTk benefiting the most using the Polarity Strategy with an improvement in F1-score of 13.7%.

The Unconstrained Strategy worked best with SEntiMoji by improving the F1-score by 7.7%. Both Lexicon Strategy and Polarity Strategy improved most with EMTk by 10.7% and 13.7% respectively. The reasons likely lie behind the feature extraction methods of EMTk, as its classification features are based on an emotion lexicon and polarity.

ESEM-E performed best with the Lexicon Strategy, outperforming the original dataset by 6.8%. As ESEM-E directly uses unigrams and bigrams as its features, it is likely that the repetition of lexical cues produced by the Lexicon Strategy significantly helped this tool. EMTk performed most effectively with the Polarity Strategy (13.7% improvement) as positive and negative sentiment polarity scores are one of its features. SEntiMoji performed best with the Polarity Strategy as well, outperforming the original dataset by 8.0%; however, SEntiMoji's performance did not vary significantly over all three augmentation approaches.

The emotions that are improved most with data augmentation strategies are Sadness and Joy, which is consistent with Murgia et al.'s [52] findings that they are easier to identify compared to other basic emotions. The reason is likely because data augmentation helped to introduce more lexical cues that were missing in the original dataset. In our analysis for RQ1, we observed that the most prevalent error category for Sadness and Joy FNs was *General Error*. Sadness achieved maximum F1-score of 0.557 in Unconstrained Strategy with SEntiMoji, and Joy achieved a maximum F1-score of 0.406 with the Polarity Strategy and ESEM-E. Surprise performed best with the Polarity Strategy where all three tools improved the F1-score with EMTk producing the best result, an F1-score of 0.630. Previous research shows that Surprise in SE is generally hard to detect, since it is not very frequent [7, 42]. However, with the addition of GoEmotions's categories and data augmentation, detection of Surprise improved significantly. As noted in previous research [7, 52], Anger and Fear are difficult to predict, as they often depend on the message context. During our error analysis, we saw that most *Implicit sentiment polarity* errors occur with Anger and Fear. These type of errors were difficult to identify even after data augmentation. SEntiMoji did not improve Anger performance in any of the strategies; while EMTk improved, its performance with the initial dataset was very low. In the case of Fear, ESEM-E did not improve with any of the strategies. However, Fear performed significantly better with the Polarity Strategy in EMTk, achieving F1-score of 0.473. Further research on what caused EMTk to perform better for Fear may help to pinpoint how to further improve classifying this emotion.

One interesting case is that with the original dataset, Love performed best across all three tools, however, with data augmentation, the performance of Love did not improve significantly. This points to a limitation of data augmentation in that it can only be of a limited benefit, i.e., useful only in cases where sufficient lexical cues are not already present in the data.

Takeaway: Overall, we observe that Data Augmentation generally improves emotion classification performance across different emotions and tools. We observe improvements especially when the initial dataset has insufficient lexical cues for a specific emotion. Out of the three augmentation strategies we experimented with,

the Polarity Strategy worked really well, as it provided a balance between completely unconstrained augmentation (which introduces noise) and highly constrained augmentation (which fails to increase size and diversity). Data augmentation is likely only able to improve performance up to point, as our current augmentation operators do not seem to help in identifying implicit emotions, such as Sarcasm.

6 RELATED WORK

Below, we describe the related work sourced from two different domains: emotion analysis of software artifacts, and data augmentation in the domain of Natural Language Processing (NLP).

6.1 Emotion Analysis of Software Artifacts

One of the earliest emotion analysis of software artifacts can be found in Murgia et al.'s work [52], where they manually analyzed 800 issue comments and concluded that some basic emotions such as Love, Joy and Sadness are easier to identify in text. In their later work, Murgia et al. [51] proposed an automated approach (namely, ESEM-E) to detect emotions in software artifacts. Motivated by their earlier work, they only focused on automatically identifying Love, Joy and Sadness in Ortu el al.'s JIRA-based dataset [61]. Calefato et al. [9] developed a feature extraction-based machine learning technique EMTk (also known as EmoTxt). They evaluated EMTk on the StackOverflow dataset initially developed by Novielli et. al. [55], which was annotated with Shaver's six basic emotions [73]. Neupane et al. [54] investigated emotion dynamics in GitHub repositories, using EMTk as their primary model. Cabrera-Diego et al. [7] used the multi-label classifiers HOMER [80] and RAkEL [81] on the above mentioned JIRA and StackOverflow datasets, observing that HOMER and RAkEL achieved a better micro F1-score than EMTk on the JIRA dataset and similar performance on the StackOverflow dataset. They concluded that EMTk is a conservative tool in general. Both the JIRA and StackOverflow dataset, however, have a limited number of utterances in certain emotion categories. The JIRA dataset only focuses on four categories of emotions (Love, Sadness, Joy, and Anger), while the StackOverflow dataset has a prevalence of Love compared to other categories and has a very limited number of Surprise instances. One of our focus areas in this paper has been on curating a dataset where all the emotion categories are represented in sufficient quantity.

Venigalla et al. [82] analyzed software developers' emotions towards software documentation using the NRC emotion analyzer module [48] of the Syuzhet package [36]. In our work, we also use the NRC's emotion module, but for data augmentation. Another thread of research in this area is based on the VAD (Valence, Arousal, and Dominance) model [68], where specific emotions are represented as a mixture of these three numerically-expressed quantities. For instance, VAD can express emotions such as, Excitement (positive valence and high arousal), Relaxation (positive valence and low arousal), Depression (negative valence and low arousal), and Stress (negative valence and high arousal). A tool named DEVA [35] was one of the first in software engineering that was based on the VAD model. Later, researchers proposed MarValous [34], which significantly outperformed DEVA (by 23.05%). Another focus of recent research has been through the use of emojis in various software artifact such as pull requests, issue comments, chats, GitHub

Table 4: Emotion classification results for all three data augmentation strategies. For F1-score, we also show the percentage improvement over the original (unaugmented) dataset.

Emotion	Strategy	Model	Precision	Recall	F1-score
Anger	Unconstrained	ESEM-E	0.567	0.250	0.347 (12.3%)
		EMTk	0.571	0.235	0.333 (66.5%)
		SEntiMoji	0.630	0.250	**0.358** (-2.5%)
	Lexicon	ESEM-E	0.581	0.265	0.364 (17.8%)
		EMTk	0.531	0.250	0.340 (70.0%)
		SEntiMoji	0.625	0.221	0.326 (-11.2%)
	Polarity	ESEM-E	0.500	0.235	0.320 (3.6%)
		EMTk	0.609	0.206	0.308 (54.0%)
		SEntiMoji	0.615	0.235	0.340 (-7.4%)
Love	Unconstrained	ESEM-E	0.596	0.636	0.615 (-4.5%)
		EMTk	0.703	0.591	0.642 (5.1%)
		SEntiMoji	0.719	0.523	0.605 (1.7%)
	Lexicon	ESEM-E	0.630	0.659	0.644 (0.0%)
		EMTk	0.659	0.614	0.635 (3.9%)
		SEntiMoji	0.710	0.500	0.587 (-1.3%)
	Polarity	ESEM-E	0.667	0.682	**0.674** (4.7%)
		EMTk	0.727	0.545	0.623 (2.0%)
		SEntiMoji	0.733	0.500	0.595 (0.0%)
Fear	Unconstrained	ESEM-E	0.545	0.150	0.235 (-19.2%)
		EMTk	0.600	0.225	0.327 (-1.8%)
		SEntiMoji	0.700	0.175	0.280 (31.5%)
	Lexicon	ESEM-E	0.600	0.150	0.231 (-20.6%)
		EMTk	0.818	0.225	0.353 (6.0%)
		SEntiMoji	0.636	0.175	0.275 (29.1%)
	Polarity	ESEM-E	0.500	0.150	0.231 (-20.6%)
		EMTk	0.867	0.325	**0.473** (42.0%)
		SEntiMoji	0.600	0.150	0.240 (12.7%)
Joy	Unconstrained	ESEM-E	0.456	0.310	0.369 (-2.4%)
		EMTk	0.486	0.214	0.298 (1.4%)
		SEntiMoji	0.477	0.250	0.328 (25.2%)
	Lexicon	ESEM-E	0.500	0.321	0.391 (3.4%)
		EMTk	0.590	0.274	0.374 (27.2%)
		SEntiMoji	0.526	0.238	0.328 (25.2%)
	Polarity	ESEM-E	0.492	0.345	**0.406** (7.4%)
		EMTk	0.613	0.226	0.330 (12.2%)
		SEntiMoji	0.575	0.274	0.371 (41.6%)
Sadness	Unconstrained	ESEM-E	0.767	0.418	0.541 (3.2%)
		EMTk	0.909	0.364	0.519 (1.4%)
		SEntiMoji	0.917	0.400	**0.557** (17.5%)
	Lexicon	ESEM-E	0.759	0.400	0.524 (0.0%)
		EMTk	0.719	0.418	0.529 (3.3%)
		SEntiMoji	0.875	0.382	0.532 (12.2%)
	Polarity	ESEM-E	0.710	0.400	0.512 (-2.3%)
		EMTk	0.821	0.418	0.554 (8.2%)
		SEntiMoji	0.913	0.382	0.538 (13.5%)
Surprise	Unconstrained	ESEM-E	0.646	0.477	0.549 (9.8%)
		EMTk	0.784	0.446	0.569 (-1.9%)
		SEntiMoji	0.767	0.354	0.484 (0.0%)
	Lexicon	ESEM-E	0.667	0.523	0.586 (17.2%)
		EMTk	0.732	0.462	0.566 (-2.4%)
		SEntiMoji	0.857	0.369	0.516 (6.6%)
	Polarity	ESEM-E	0.654	0.523	0.581 (16.2%)
		EMTk	0.791	0.523	**0.630** (8.6%)
		SEntiMoji	0.852	0.354	0.500 (3.3%)
Overall	Unconstrained	ESEM-E	0.587	0.368	0.453 (3.0%)
		EMTk	0.659	0.326	0.436 (3.3%)
		SEntiMoji	0.681	0.317	0.433 (7.7%)
		Average	0.642	0.337	0.441 (4.8%)
	Lexicon	ESEM-E	0.610	0.382	0.470 (6.8%)
		EMTk	0.658	0.362	0.467 (10.7%)
		SEntiMoji	0.699	0.306	0.426 (6.0%)
		Average	0.656	0.350	0.454 (7.8%)
	Polarity	ESEM-E	0.593	0.385	0.467 (6.1%)
		EMTk	0.734	0.357	**0.480** (13.7%)
		SEntiMoji	0.712	0.312	0.434 (8.0%)
		Average	0.680	0.351	0.460 (9.3%)

discussions, etc. [10, 12, 43, 67, 76]. Chen el al. proposed a technique called SEntiMoji that uses emojis to recognize emotions [13, 14], based on an existing emoji representation model DeepMoji [23]. SEntiMoji is able to produce output in Shaver's categories or in the VAD model. When SEntiMoji was compared against EMSE-E [51] and EMTk [9] on the JIRA and StackOverflow dataset, it showed overall improvement across all six basic emotions. SEntiMoji was also compared against DEVA [35] and MarValous [34], and was shown to achieve better performance than these two tools as well. In our research, we compare the tools that use Shaver's emotion categories, SEntiMoji, EMSM-E, and EMTk, using a GitHub-based dataset that we curate. We consider data augmentation as the means to improve performance in all three of these tools.

6.2 Data Augmentation in NLP

The augmentation of textual data (i.e., in NLP) has been an area of considerable interest in recent years to address the data scarcity problem related to different tasks such as, subjectivity detection [32], question understanding and summarization [49]. Early techniques of interest in data augmentation included synonym replacement [89], i.e., replacing a word with its synonym, and BackTranslation [72], i.e., paraphrasing a text by converting to a second language and then converting back to original language. Inspired by similar approaches in computer vision, researchers also devised MixUp augmentation [74, 88], which meshes together existing examples in order to create new augmented instances. Another recent research direction is Unsupervised Data Augmentation [86] (UDA) which is a semi-supervised model that outperformed state-of-the-art methods using only 20 labeled instances. Wei el al. [83] proposed a method - Easy Data Augmentation (EDA) which worked surprisingly well for smaller datasets despite being a simple technique that uses straightforward operators, e.g., synonym replacement using WordNet [47], random insertion, random swap, and random deletion. As software artifact datasets are often small [64], we were inspired by EDA in devising our data augmentation technique. A recent research thread in data augmentation is contextual augmentation [74], using various large language models such as CBERT [85], BART [38], GPT-2 [2], etc. Kumer at al. [38] showed that for classification tasks, BART outperforms other models due to its ability to generate longer sequences of text in context. In this paper, we also leverage the BART model as part of our augmentation operators.

7 THREATS TO VALIDITY

Several limitations may impact the interpretation of our findings. We categorize and list each of them below.

Construct validity. Construct validity concerns the relationship between theory and observation. Shaver's emotions model [73] and the GoEmotions model by Demszky et al. [19] are two different schema. We use a combination of the two, which may violate their original design. To mitigate this risk, we carefully read both of the original researches and used Shaver's model as our primary schema, integrating GoEmotions' categories only when they are complementary and do not conflict with Shaver's in any part of their definitions. Furthermore, the error analysis in RQ1 shows that none of the emotion categories that integrated GoEmotions are among the worse performing. Instead, the addition of GoEmotions secondary emotion categories specifically improves the performance of the basic category Surprise, which has exhibited relatively low F1-score in previous research in software engineering emotion classification [42].

Internal validity. Internal validity concerns the study design factors that may influence the results. One such threat to our study is not doing cross validation, which would have improved the reliability of the results. We mitigate this threat by using stratified sampling and a reasonable train-test data split of 80%-20% respectively. Another threat is that, to conduct our experiments we use existing tools and their released code, except for ESEM-E [51]. It is possible that we have incorrectly implemented ESEM-E, although, we explicitly followed the authors' instructions to mitigate this threat. The subjectivity of annotating emotions (and in the error analysis) presents another threat to internal validity. However, the use of a three tiered emotion structure and high inter rater agreement (> 0.8) ensure the reliability of the annotation procedure.

External validity. External validity concerns the generalization of our findings. Our study shows that data augmentation improves emotion classification across the three tools we experimented with. However, the specific augmentation strategies may not generalize beyond the three tools we studied and our dataset extracted from GitHub comments. More specifically, our findings may not generalize over other types of artifacts in software engineering, such as StackOverflow, JIRA, etc. While our results introduce the potential of data augmentation for emotion classification, further investigation is needed in other to validate our results beyond the tools and the data used in our study.

8 CONCLUSION AND FUTURE WORK

In this paper, we first conduct a qualitative study to understand the limitations of the existing machine learning-based tools for classifying emotions in software engineering-related text. Specifically, we evaluate ESEM-E [51], EMTk [9], SEntiMoji [13] on our curated dataset of 2000 GitHub pull requests and issue comments. We observe that some types of errors could be more difficult to address than others, however there is a scope to improve the performance of the existing tools by creating better training data. Thus, next we investigate three types of data augmentation strategies that could be leveraged to improve emotion detection in software-related text. Our results indicate that augmentation operators that target words with specific polarity are significantly more effective than generic augmentation operators. Using polarity-based augmentation shows an average improvement of 9.3% in micro F1-Score across the three existing emotion classification tools.

Our immediate next steps focus on investigating more diverse data sources, including other types of software artifacts such as StackOverflow, Slack chats, etc. We are also interested in exploring new data augmentation techniques based on large language models that are pre-trained on software engineering corpora and fine-tuned to emotion and sentiment-type tasks. Finally, we would like to explore if polarity based data augmentation strategies could improve other related tasks in software engineering, such as sentiment analysis.

REFERENCES

[1] T. Amabile, Sigal G. Barsade, J. S. Mueller, and B. Staw. 2005. Affect and Creativity at Work. *Administrative Science Quarterly* 50 (2005), 367 – 403.

[2] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, N. Tepper, and Naama Zwerdling. 2020. Do Not Have Enough Data? Deep Learning to the Rescue!. In *AAAI*.

[3] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.

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

[5] Zdravko Botev and Ad Ridder. 2017. Variance reduction. *Wiley statsRef: Statistics reference online* (2017), 1–6.

[6] Mohamad Adam Bujang and Nurakmal Baharum. 2017. A Simplified Guide to Determination of Sample Size Requirements for Estimating the Value of Intraclass Correlation Coefficient: a Review. *Archives of Orofacial Science* 12, 1 (2017).

[7] Luis Adrián Cabrera-Diego, Nik Bessis, and Ioannis Korkontzelos. 2020. Classifying emotions in Stack Overflow and JIRA using a multi-label approach. *Knowledge-Based Systems* 195 (2020), 105633.

[8] Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. 2017. Sentiment Polarity Detection for Software Development. *Empirical Software Engineering* 23 (2017), 1352–1382.

[9] Fabio Calefato, Filippo Lanubile, Nicole Novielli, and Luigi Quaranta. 2019. EMTk - The Emotion Mining Toolkit. In *2019 IEEE/ACM 4th International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. 34–37.

[10] P. Chatterjee, K. Damevski, N.A. Kraft, and L. Pollock. 2020. Automatically Identifying the Quality of Developer Chats for Post Hoc Use. In *Transactions on Software Engineering and Methodology (TOSEM) (TOSEM '20)*.

[11] Preetha Chatterjee, Kostadin Damevski, and Lori Pollock. 2021. Automatic Extraction of Opinion-based Q&A from Online Developer Chats. In *Proceedings of the 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. 1260–1272. https://doi.org/10.1109/ICSE43902.2021.00115

[12] P. Chatterjee, K. Damevski, L. Pollock, V. Augustine, and N.A. Kraft. 2019. Exploratory Study of Slack Q&A Chats as a Mining Source for Software Engineering Tools. In *Proceedings of the 16th International Conference on Mining Software Repositories (MSR'19)* (Montreal, Canada). https://doi.org/10.1109/MSR.2019.00075

[13] Zhenpeng Chen, Yanbin Cao, Xuan Lu, Qiaozhu Mei, and Xuanzhe Liu. 2019. SEntiMoji: An Emoji-Powered Learning Approach for Sentiment Analysis in Software Engineering. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Tallinn, Estonia) *(ESEC/FSE 2019)*. Association for Computing Machinery, New York, NY, USA, 841–852. https://doi.org/10.1145/3338906.3338977

[14] Zhenpeng Chen, Yanbin Cao, Huihan Yao, Xuan Lu, Xin Peng, Hong Mei, and Xuanzhe Liu. 2021. Emoji-Powered Sentiment and Emotion Detection from Software Developers' Communication Data. *ACM Trans. Softw. Eng. Methodol.* 30, 2, Article 18 (Jan. 2021), 48 pages. https://doi.org/10.1145/3424308

[15] Alan S Cowen, Hillary Anger Elfenbein, Petri Laukka, and Dacher Keltner. 2019. Mapping 24 emotions conveyed by brief human vocalization. *American Psychologist* 74, 6 (2019), 698.

[16] Alan S. Cowen and Dacher Keltner. 2017. Self-report captures 27 distinct categories of emotion bridged by continuous gradients. 114, 38 (2017), E7900–E7909.

[17] Alan S Cowen and Dacher Keltner. 2020. What the face displays: Mapping 28 emotions conveyed by naturalistic expression. *American Psychologist* (2020).

[18] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. 2018. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine* 35, 1 (2018), 53–65. https://doi.org/10.1109/MSP.2017.2765202

[19] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan S. Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 4040–4054. https://doi.org/10.18653/v1/2020.acl-main.372

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186.

[21] Felipe Ebert, Fernando Castor, Nicole Novielli, and Alexander Serebrenik. 2021. An exploratory study on confusion in code reviews. *Empirical Software Engineering* 26 (2021), 1–48.

[22] Paul Ekman. 1999. Basic Emotions. In *Handbook of Cognition and Emotion*, Tim Dalgleish and M. J. Powers (Eds.). Wiley, 4–5.

[23] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1615–1625.

[24] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A Survey of Data Augmentation Approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 968–988.

[25] Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler. 2021. The SPACE of Developer Productivity: There's More to It than You Think. 19, 1 (2021).

[26] G. Fucci, N. Cassee, F. Zampetti, N. Novielli, A. Serebrenik, and M. Di Penta. 2021. Waiting Around or job Half-Done? Sentiment in Self-Admitted Technical Debt. In *2021 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR) (MSR)*. IEEE Computer Society, Los Alamitos, CA, USA, 403–414. https://doi.org/10.1109/MSR52588.2021.00052

[27] Daniela Girardi, Nicole Novielli, Davide Fucci, and Filippo Lanubile. 2020. Recognizing Developers' Emotions While Programming. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (Seoul, South Korea) *(ICSE '20)*. Association for Computing Machinery, New York, NY, USA, 666–677.

[28] Daniel Graziotin, Fabian Fagerholm, Xiaofeng Wang, and Pekka Abrahamsson. 2017. On the Unhappiness of Software Developers. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering* (Karlskrona, Sweden) *(EASE'17)*. Association for Computing Machinery, New York, NY, USA, 324–333. https://doi.org/10.1145/3084226.3084242

[29] Daniel Graziotin, Fabian Fagerholm, Xiaofeng Wang, and Pekka Abrahamsson. 2018. What happens when software developers are (un)happy. *Journal of Systems and Software* 140 (2018), 32–47. https://doi.org/10.1016/j.jss.2018.02.041

[30] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. 2013. Are Happy Developers More Productive?. In *Product-Focused Software Process Improvement*, Jens Heidrich, Markku Oivo, Andreas Jedlitschka, and Maria Teresa Baldassarre (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 50–64.

[31] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. 2015. Do Feelings Matter? On the Correlation of Affects and the Self-Assessed Productivity in Software Engineering. 27, 7 (2015).

[32] Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting Data with Mixup for Sentence Classification: An Empirical Study. *ArXiv* abs/1905.08941 (2019).

[33] 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. Association for Computing Machinery, New York, NY, USA.

[34] Md Rakibul Islam, Md Kauser Ahmmed, and Minhaz F. Zibran. 2019. MarValous: Machine Learning Based Detection of Emotions in the Valence-Arousal Space in Software Engineering Text. Association for Computing Machinery, New York, NY, USA.

[35] Md Rakibul Islam and Minhaz F. Zibran. 2018. DEVA: Sensing Emotions in the Valence Arousal Space in Software Engineering Text. Association for Computing Machinery, New York, NY, USA.

[36] Matthew L. Jockers. 2015. *Syuzhet: Extract Sentiment and Plot Arcs from Text*. https://github.com/mjockers/syuzhet

[37] Venelin Kovatchev, Phillip Smith, Mark G. Lee, and Rory T. Devine. 2021. Can vectors read minds better than experts? Comparing data augmentation strategies for the automated scoring of children's mindreading ability. In *ACL*.

[38] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data Augmentation using Pre-trained Transformer Models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*. Association for Computational Linguistics, Suzhou, China, 18–26.

[39] Miikka Kuutila, M. Mäntylä, and Maëlick Claes. 2020. Chat activity is a better predictor than chat sentiment on software developers productivity. *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops* (2020).

[40] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7871–7880.

[41] Bohan Li, Yutai Hou, and Wanxiang Che. 2022. Data augmentation approaches in natural language processing: A survey. *AI Open* (2022).

[42] Bin Lin, Nathan Cassee, Alexander Serebrenik, Gabriele Bavota, Nicole Novielli, and Michele Lanza. 2022. Opinion Mining for Software Development: A Systematic Literature Review. *ACM Trans. Softw. Eng. Methodol.* 31, 3, Article 38 (mar 2022), 41 pages.

[43] Xuan Lu, Yanbin Cao, Zhenpeng Chen, and Xuanzhe Liu. 2018. A first look at emoji usage on github: An empirical study. *arXiv preprint arXiv:1812.04863* (2018).

[44] Edward Ma. 2019. NLP Augmentation. https://github.com/makcedward/nlpaug.

[45] Mika V Mäntylä, Nicole Novielli, Filippo Lanubile, Maëlick Claes, and Miikka Kuutila. 2017. Bootstrapping a lexicon for emotional arousal in software engineering. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*. IEEE, 198–202.

[46] André N. Meyer, Earl T. Barr, Christian Bird, and Thomas Zimmermann. 2021. Today Was a Good Day: The Daily Life of Software Developers. *IEEE Transactions on Software Engineering* 47, 5 (2021), 863–880.

[47] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[48] Saif M Mohammad and Peter D Turney. 2013. Nrc emotion lexicon. *National Research Council, Canada* 2 (2013).

[49] Khalil Mrini, Franck Dernoncourt, Seunghyun Yoon, Trung Bui, Walter Chang, Emilia Farcas, and Ndapa Nakashole. 2021. A Gradually Soft Multi-Task and Data-Augmented Approach to Medical Question Understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 1505–1515.

[50] Sebastian C. Müller and Thomas Fritz. 2015. Stuck and Frustrated or in Flow and Happy: Sensing Developers' Emotions and Progress. In *Proceedings of the 37th International Conference on Software Engineering - Volume 1* (Florence, Italy) *(ICSE '15)*. IEEE Press, 688–699.

[51] 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 Softw. Engg.* 23, 1 (feb 2018), 521–564. https://doi.org/10.1007/s10664-017-9526-0

[52] A. Murgia, Parastou Tourani, B. Adams, and Marco Ortu. 2014. Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In *MSR 2014*.

[53] 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

[54] Krishna Prasad Neupane, Kabo Cheung, and Yi Wang. 2019. EmoD: An end-to-end approach for investigating emotion dynamics in software development. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 252–256.

[55] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. 2018. A gold standard for emotion annotation in stack overflow. In *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*. IEEE, 14–17.

[56] Nicole Novielli, Fabio Calefato, Filippo Lanubile, and Alexander Serebrenik. 2021. Assessment of off-the-shelf SE-specific sentiment analysis tools: An extended replication study. *Empirical Software Engineering* 26, 4 (Jun 2021). https://doi.org/10.1007/s10664-021-09960-w

[57] 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.

[58] 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)*. IEEE, 364–375.

[59] Nicole Novielli and Alexander Serebrenik. 2019. Sentiment and Emotion in Software Engineering. *IEEE Software* 36, 5 (2019), 6–23.

[60] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli. 2015. Are Bullies More Productive? Empirical Study of Affectiveness vs. Issue Fixing Time. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. 303–313.

[61] Marco Ortu, Alessandro Murgia, Giuseppe Destefanis, Parastou Tourani, Roberto Tonelli, Michele Marchesi, and Bram Adams. 2016. The Emotional Side of Software Developers in JIRA. In *Proceedings of the 13th International Conference on Mining Software Repositories* (Austin, Texas) *(MSR '16)*. Association for Computing Machinery, New York, NY, USA, 480–483. https://doi.org/10.1145/2901739.2903505

[62] W Gerrod Parrott. 2001. *Emotions in social psychology: Essential readings*. psychology press.

[63] Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. In *Theories of emotion*. Elsevier, 3–33.

[64] Julian Aron Aron Prenner and Romain Robbes. 2021. Making the most of small Software Engineering datasets with modern machine learning. *IEEE Transactions on Software Engineering* (2021), 1–1.

[65] Husam Quteineh, Spyridon Samothrakis, and Richard Sutcliffe. 2020. Textual data augmentation for efficient active learning on tiny datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 7400–7410.

[66] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[67] Shiyue Rong, Weisheng Wang, Umme Ayda Mannan, Eduardo Santana de Almeida, Shurui Zhou, and Iftekhar Ahmed. 2022. An empirical study of emoji use in software development communication. *Information and Software Technology* (2022), 106912.

[68] James A Russell and Albert Mehrabian. 1977. Evidence for a three-factor theory of emotions. *Journal of research in Personality* 11, 3 (1977), 273–294.

[69] Arghavan Sanei, Jinghui Cheng, and Bram Adams. 2021. The Impacts of Sentiments and Tones in Community-Generated Issue Discussions. In *2021 IEEE/ACM 13th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 1–10.

[70] Farhana Sarker, Bogdan Vasilescu, Kelly Blincoe, and Vladimir Filkov. 2019. Socio-Technical Work-Rate Increase Associates with Changes in Work Patterns in Online Projects. IEEE Press.

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

[72] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 86–96.

[73] Phillip R. Shaver, Judith C. Schwartz, Donald Kirson, and Cary O'Connor. 1987. Emotion knowledge: further exploration of a prototype approach. *Journal of personality and social psychology* 52 6 (1987), 1061–86.

[74] Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data* 8, 1 (2021), 1–34.

[75] V. Sinha, A. Lazar, and B. Sharif. 2016. Analyzing Developer Sentiment in Commit Logs. In *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*. 520–523.

[76] Teyon Son, Tao Xiao, Dong Wang, Raula Gaikovina Kula, Takashi Ishio, and Kenichi Matsumoto. 2021. More Than React: Investigating The Role of EmojiReaction in GitHub Pull Requests. *arXiv preprint arXiv:2108.08094* (2021).

[77] Steven E Stemler. 2004. A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. (2004).

[78] Jeniya Tabassum, Mounica Maddela, Wei Xu, and Alan Ritter. 2020. Code and Named Entity Recognition in StackOverflow. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online.

[79] 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.

[80] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2008. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, Vol. 21.

[81] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2010. Random k-labelsets for multilabel classification. *IEEE transactions on knowledge and data engineering* 23, 7 (2010), 1079–1089.

[82] Akhila Sri Manasa Venigalla and Sridhar Chimalakonda. 2021. Understanding Emotions of Developer Community Towards Software Documentation. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. 87–91.

[83] Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 6382–6388.

[84] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45.

[85] Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International Conference on Computational Science*. Springer, 84–95.

[86] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems* 33 (2020), 6256–6268.

[87] Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen Mckeown, Ramesh Nallapati, Andrew O Arnold, and Bing Xiang. 2021. Supporting Clustering with Contrastive Learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 5419–5430.

[88] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[89] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems* 28 (2015).