A Four-Dimension Gold Standard Dataset for Opinion Mining in Software Engineering

Md Rakibul Islam[1], Md Fazle Rabbi[2], Youngeun Jo[1], Arifa Champa[2], Ethan Young[1], Camden Wilson[1], Gavin Scott[1], Minhaz Zibran[2]

[1] Lamar University, {mislam108, jyoungeun, eyoung12, cwilson65, gscott7}@lamar.edu

[2] Idaho State University, {mdfazlerabbi, arifaislamchampa, minhazzibran}@isu.edu

ABSTRACT

We present the first four-dimension gold standard dataset to advance opinion mining focused on the software engineering domain. Through a well-defined sampling and annotation strategy leveraging multiple coders, we construct a corpus of 2,000 Stack Overflow posts labeled with four dimensions/tuples, including sentiments, polar facts, aspects, and named entities. This multidimensional ground truth dataset opens up new research opportunities for opinion mining in domain-adapted NLP tools for software engineering by capturing existing relationships between extracted elements at a more granular level. It also facilitates investigating the effects of sentiments in the developers' social forums.

CCS CONCEPTS

• **Software and its engineering** → *Programming teams.*

KEYWORDS

Sentiment Analysis, Opinion Mining, Aspects, Named Entity, Software Engineering, Natural Language Processing

ACM Reference Format:
Md Rakibul Islam[1], Md Fazle Rabbi[2], Youngeun Jo[1], Arifa Champa[2], Ethan Young[1], Camden Wilson[1], Gavin Scott[1], Minhaz Zibran[2]. 2024. A Four-Dimension Gold Standard Dataset for Opinion Mining in Software Engineering. In *21st International Conference on Mining Software Repositories (MSR '24), April 15–16, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3643991.3644893

1 INTRODUCTION

Sentiment analysis and opinion mining in software engineering texts such as developer forums, issue trackers, and code reviews have become an increasingly important research area [1, 5, 9, 11, 17, 21, 25]. Understanding sentiments, opinions, and emotions expressed in these texts can lead to data-driven insights about barriers faced in collaborative software development and enhanced tools leveraging natural language processing (NLP) techniques.

A few datasets are constructed for sentiment analysis and opinion mining in software engineering. While most of the existing datasets focus only on overall sentiment polarities and emotions [1, 2, 21, 23], there are only two datasets available that extract aspects along with sentiments. Moreover, the latter two datasets are limited at the sentence level [18, 25]. There is also a scarcity of named entity annotated datasets that hinder the analysis of named entities in software engineering. This lack of the required dataset limits studying nuanced relationships between extracted named entities and their aspects and sentiments, which is essential for opinion mining in software engineering.

This work creates and releases to the public [15] a new richly annotated dataset for opinion mining in software engineering constructed from Stack Overflow posts. We employ an opportunistic sampling strategy to obtain 2,000 posts expressing sentiments and opinions about software-specific named entities. Using a rigorous annotation process, these texts are annotated along four dimensions, i.e., tuples that include sentiments, polar facts, aspects, and named entities. This multidimensional ground truth dataset is first of its type in software engineering that enables novel research directions for domain-adapted NLP while supporting fundamental studies about the role of sentiments in collaborative software knowledge-sharing forums, such as Stack Overflow.

2 BACKGROUND

Opinion: An opinion is a statement that conveys a subjective view or judgment about named entities. Bing Liu [20] defined opinion as: "An opinion is a quintuple $< e_i, a_{ij}, s_{ijkl}, h_k, t_l >$, where e_i is the name of the entity, a_{ij} is an aspect of e_i, s_{ijkl} is the sentiment on aspect a_{ij} of entity e_i, h_k is the opinion holder, and t_l is the time when the opinion is expressed by h_k."

In the following, we define sentiment, aspect, and named entity, along with polar facts that we annotated in our dataset.

Sentiment. Sentiment refers to the emotions or feelings that are conveyed through written or spoken language. More specifically, sentiment analysis examines a text to determine the overall positive, negative, or neutral attitude. For example, "Hurrah! I fixed the bug" expresses a positive sentiment, and the sentence "The bug is making my life a hell" expresses a negative sentiment.

Polar fact. Polar facts are statements or phrases that describe positive or negative factual information about something without conveying sentiments. The sentence "The bug is fixed, and the system works" is a positive fact in software engineering. Again, the sentence "The deployed patch is not working." is an example of a negative fact.

In this data annotation process, we decided to annotate the polar facts for two reasons: (i) annotators often mix up polar facts with sentiments that result in unreliable annotation of sentiments [18, 22]. Thus, we kept sentiments and polar facts separate for more reliable annotation, and (ii) polar facts can provide useful information on many occasions.

MSR '24, April 15–16, 2024, Lisbon, Portugal
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0587-8/24/04.
https://doi.org/10.1145/3643991.3644893

Aspects. Developers discussed and shared opinions on various attributes and properties of named entities in the forum posts. Some of the key aspects that came up frequently included:

Performance - related to the named entities' speed, scalability, efficiency, and other performance characteristics. *Usability* - covering how easy or difficult it is to use the named entities effectively. *Security* - discussing the vulnerability, encryption, access control, and other security-related features supported by the named entities. *Documentation* - feedback on the availability and quality of official and informal named entities' documentation resources. *Functionality* - the features offered/not offered by the named entities. *Learnability* - covering how easy or difficult it is to learn the named entities. *Compatibility* - dealing with the ability of the named entities to function with specific frameworks, platforms, or other components. *Portability* - related to the ability to compile and run the named entities across different operating systems and computing platforms. *Community* - concerning the availability and responsiveness of forums, mailing lists, and contributors assisting users of named entities. *Legal* - opinions around the named entities' licensing terms, pricing models, and permissible usage contexts. *Bugs* - discussing and identifying defects, errors, and crashes in the named entities. *Popularity* - related to the named entities' popularity. *Reliability* - refers to a system or software's consistent and dependable performance under specific conditions over time. *Usefulness* - pertains to the practical value or effectiveness of a system, product, or information in meeting particular needs or objectives.

We also assign *simple info/others* as an aspect to any textual item related to a named entity that does not fall into the aspect categories mentioned above.

Named entity. A named entity refers to real-world objects, individuals, locations, organizations, quantities, dates, or expressions with specific names. In software engineering, the names of a programming language, e.g., Java, and a framework, e.g., ASP.NET, are examples of two named entities.

Yang et al. [26] prepared a gazetteer for the software engineering domain that includes 400,147 entries divided into five named entity categories, such as programming language (Pl), platform (Plat), Application Programming Interface (API), tool-library-framework (Fram), and software standard (Stan). Table 1 presents the distribution of the named entities in the gazetteer according to those five categories with examples.

3 DATASET CONSTRUCTION

3.1 Collecting Samples for Annotation

We downloaded the Stack Overflow data dump released on Jun 6th, 2022. We collected the text items that included answers and comments associated with posts from the dump. We excluded questions as developers expressed their opinions in the answers and comments [2, 17]. To improve the readability of the texts, we preprocessed each text item to discard code snippets, URLs, and HTML tags using regular expressions.

Finding a text item containing at least one named entity with sentiment can be compared to looking for a needle in a haystack, as the data dump includes millions of sentences with no entity and sentiment. To overcome this challenge, we used an opportunistic sampling strategy. Initially, we used a dictionary of a popular domain-independent sentiment analysis tool, `SentiStrength` [24], to assess the presence/absence of a sentimental lexicon (e.g., good, love, and harmful) in the text items. We selected those text items that had at least one sentimental word in them.

Table 1: Categories and Number of Software-specific Entities

Named Entity Category	# Entries
Programming language (e.g., Java, C)	419
Platform (e.g., x86, AMD64)	175
API (e.g., Java ArrayList, toString())	396,968
Tool-library-framework (e.g., Eclipse)	2,196
Software standard (e.g., HTTP, FTP)	389

Then, we applied the following strategy to ensure each post had at least one named entity. We randomly selected 50 entries from each category of the gazetteer [26] described in Section 2. Then, we searched each selected named entity in the posts, i.e., answers and comments. We converted named entities, answers, and comments into lowercase strings. Then, we searched each selected named entity in the posts. We selected those posts that had a match.

Finally, we randomly selected 2,000 posts for manual annotation by human annotators. While relevant to the topics of interest, many Stack Overflow posts selected through our sampling strategy contained extensive content (with more than ten sentences) unsuitable for inclusion in the gold standard dataset. Therefore, we developed a protocol to identify and extract only the most informative sentences from each post for analysis and potential dataset inclusion. The first two authors of the paper jointly read through each selected post and flagged individual sentences that exhibited relevance to our interest. Only these manually flagged sentences - typically one to three per post - were considered when determining inclusion in the final gold standard dataset. This targeted extraction process enabled the efficient distillation of lengthy Stack Overflow posts, resulting in a high-quality, fit-for-purpose gold dataset.

3.2 Annotating Collected Samples

3.2.1 Human annotators.

A total of six annotators participated in the annotation task. Among the six annotators, two were Ph.D. students, one was doing post-baccalaureate degree, and the remaining three were senior undergraduate students. All the annotators were pursuing their computer science degrees at two universities in North America. While the first three students had at least one year of professional experience in software engineering, the senior students had experience in doing internships in two different software companies.

3.2.2 Training the human annotators.

We first conducted training in the following three phases to start the annotation.

Phase-1: Common understanding. Here, the annotators were first provided definitions and discussion of each sentiment, aspect, polar fact, and named entity type, along with five carefully selected examples of each. The sentiment examples were drawn from the dataset compiled by Ortu et al. [23], the aspect examples from the datasets by Uddin and Khomh [25], and the named entity examples from the dataset by Ye et al. [26]. Although no dataset was annotated with polar facts, we collected four examples of polar facts from the discussions mentioned elsewhere [21, 22].

These initial examples helped establish a common understanding of the annotation categories and guidelines before beginning the annotation process. Ensuring clarity on the relevant classifications and types was an essential first step in achieving high inter-annotator agreement and consistent application of codes throughout the annotation task. This phase was two hours long and conducted by the first author.

Phase-2: Annotation jointly. The 2,000 text items were divided into three groups to facilitate annotation. Group 1 and Group 2 contained 100 texts each and were used to train the annotators, while the remaining 1,800 texts were assigned to Group 3 to complete annotation. To begin, all annotators were assigned the same 100 texts from Group 1 to annotate collaboratively under the supervision of the first author. By annotating this initial set of texts together, the goal was to align the annotators and minimize individual biases. This phase was 60 minutes long.

Phase-3: Annotation individually. Next, the annotators were given the 100 unique texts in Group 2 to annotate independently. The annotators were given eight hours to complete their annotations. Upon completion, inter-annotator conflicts were identified through comparison. The annotators and the first author then discussed these disagreements as a group to resolve annotation differences. This allowed for identifying potential gaps in understanding and, ultimately, convergence on a common ground for coding texts.

With a shared understanding established from Groups 1 and 2, the annotators were prepared to move on to robust annotation of the larger 1,800 texts in Group 3.

3.2.3 Completing the annotation. At this point, the annotators were divided into two teams. Each team was assigned 900 text items from the remaining 1,800 text items in Group-3. This time, the text items were uploaded to the web-based `Inception` [16] tool for facilitating the annotation process.

After completing the annotation, the gold label was obtained by applying majority voting. We also measured the reliability of the annotations by computing the Fleiss' Kappa [4] values of the agreements between the annotators. The Fleiss' Kappa values for each type of sentiment, polar fact, aspect, and named entity are mentioned in Table 2. The Fleiss' Kappa values range between 0.42 and 0.71 for each dimension, indicating that the agreements between the annotators were substantially high, ensuring the annotations' high reliability. The obtained Fleiss' Kappa values are comparable to those observed by previous annotations performed elsewhere [2, 23]. Despite the high agreement between the annotators, in a few cases, they showed disagreement where the first author discussed with the annotators to reach an agreement and assigned the final labels.

4 DATASET DESCRIPTION

To give a high-level overview, in Table 2, we present the results of the annotation process on our dataset, demonstrating the distribution or frequency (n), percentage (ρ) and rater agreement (Fleiss' Kappa (k)) for various dimensions. For example, there are 396 sentences with Positive (*Pos*) sentiment, which makes up 19.8% of all the sentences, and a k score of 0.64. The percentages presented herein may not add up to 100% due to the allowance for multiple labeling of sentences. In our dataset, sentences with neutral sentiment are most frequent (n=1,515), while for polar facts, positive statements are most frequent (n=868). In aspect, sentences describing *Functionality* are the most frequent (n=533), while for named entities, *Framework* has the highest frequency (n=1548).

To give an in-depth breakdown, Table 3 shows the distribution of sentiments (Se) and polar facts (Po) across aspects for each named entity. We identify the sentiment and polar fact for each aspect and named entity pair in a sentence. In the table, P, N, and Ne denote positive, negative, neutral sentiment, or polar fact, respectively.

Table 2: Distribution and Kappa Value for Dimensions/Tuples

Tuple	Distribution (n), Percentage (ρ), Fleiss' Kappa (k)
Sentiment	Pos (n=396, ρ=19.8%, k=0.64); Neg (n=106, ρ=5.15%, k=0.51); Neu (n=1,515, ρ=75.75%, k=0.71)
Polar Facts	Pos (n=868, ρ=43.4%, k=0.70); Neg (n=278, ρ=13.9%, k=0.58); Neu (n=897, ρ=44.85%, k=0.67)
Aspect	Bug (n=15, ρ=0.75%, k=0.44); Community (n=64, ρ=3.2%, k=0.46); Compatibility (n=217, ρ=10.85%, k=0.67); Documentation (n=70, ρ=3.5%, k=0.50); Functionality (n=533, ρ=26.65%, k=0.68); Learnability (n=100, ρ=5%, k=0.48); Legal (n=5, ρ=0.25%, k=0.42); Performance (n=231, ρ=11.55%, k=0.58); Popularity (n=66, ρ=3.3%, k=0.60); Portability (n=40, ρ=2%, k=0.47); Reliability (n=68, ρ=3.4%, k=0.57); Security (n=22, ρ=1.1%, k=0.49); Usability (n=331, ρ=16.55%, k=0.68); Usefulness (n=82, ρ=4.1%, k=0.67); Simple info/Others (n=506, ρ=25.3%, k=0.67)
Named Entity	PL (n=339, ρ=16.95%, k=0.67); Plat (n=4.45, ρ=19.8%, k=0.58); Fram (n=1,548, ρ=77.4%, k=0.68); Stan (n=99, ρ=4.95%, k=0.59)

Table 3: Sentiments and polar facts for aspects across entities

Named Entity →		**Pl**			**Plat**			**API**			**Fram**			**Stan**		
Aspect ↓		P	N	Ne	P	N	Ne	P	N	Ne	P	N	Ne	P	N	Ne
Bug	Se	0	0	0	0	0	0	0	0	2	2	2	9	0	0	1
	Po	0	0	0	0	0	0	0	0	2	4	3	6	0	0	1
Community	Se	3	1	11	1	0	2	0	0	2	17	6	36	1	0	1
	Po	8	4	5	3	0	0	1	0	1	30	12	16	2	0	0
Compatibility	Se	12	3	34	2	0	16	1	2	24	33	5	147	5	1	7
	Po	21	7	21	8	2	8	7	7	14	74	33	79	6	4	3
Documentation	Se	0	0	2	0	1	0	0	0	10	2	6	22	0	0	1
	Po	0	0	2	0	1	0	0	0	10	10	13	9	0	0	1
Functionality	Se	22	2	59	1	0	6	9	6	126	84	28	294	5	1	26
	Po	41	10	38	1	1	5	24	21	98	192	52	169	16	1	15
Learnability	Se	8	3	24	1	0	5	2	0	3	18	2	56	1	1	1
	Po	12	8	14	4	0	2	3	0	2	38	12	28	2	1	0
Legal	Se	0	0	1	0	0	1	0	0	0	5	0	0	0	0	0
	Po	1	0	0	0	0	1	0	0	0	3	0	2	0	0	0
Performance	Se	6	2	18	1	0	12	6	1	25	36	12	135	4	1	13
	Po	15	4	10	4	6	4	12	6	15	83	26	79	8	1	8
Popularity	Se	5	1	12	0	0	6	1	0	5	9	2	46	1	0	1
	Po	8	1	9	2	0	4	3	0	3	28	3	26	1	0	1
Portability	Se	4	1	6	1	0	3	0	0	3	7	1	21	1	0	1
	Po	7	1	3	4	0	0	1	0	2	18	5	6	1	0	1
Reliability	Se	4	1	5	1	0	3	1	0	3	19	6	31	0	2	4
	Po	6	1	3	3	1	0	1	0	3	29	11	16	0	2	4
Security	Se	0	0	1	0	0	0	0	0	2	0	1	14	0	0	4
	Po	0	0	1	0	0	0	1	0	1	5	2	9	0	0	4
Usability	Se	12	4	39	2	0	9	6	2	34	66	21	178	2	2	12
	Po	33	9	15	3	1	6	17	6	21	142	40	85	6	2	8
Usefulness	Se	3	0	10	1	0	4	1	3	11	20	2	41	2	0	1
	Po	7	0	6	4	0	1	6	5	4	34	4	25	2	0	1
Simple info / Others	Se	19	4	54	5	0	17	6	2	100	76	16	296	4	1	17
	Po	35	7	36	13	2	8	19	4	85	153	40	201	11	4	9

Favorable discussions about the named entity *Fram* are found across aspects related to *Compatibility, Functionality, Performance, Usability, and Others.* In *Functionality*, it is observed that neutral sentiments are commonly expressed across all named entities, and this trend remains consistent across all aspects. Notably, *legal*, *bug*, and *security* aspects are the least discussed across all named entities.

5 RESEARCH OPPORTUNITIES

The gold standard dataset constructed in this work opens up several exciting research opportunities. One such area is evaluating the performance of aspect-based sentiment analysis models on software engineering domain-specific text data. This dataset provides the necessary ground truth annotations for aspect terms, associated sentiments, polar facts, and entities, which can serve as test data for new models and algorithms. Comparative assessment against existing models trained on generic reviews can reveal domain-specific performance gaps to guide further research.

Another opportunity is using this dataset to train aspect extraction and sentiment classification systems tailored to the software engineering domain. The overall goal would be improving performance on software-specific tasks, such as analyzing user feedback in app reviews or prioritizing bug reports. Given the granular aspect, sentiment, and textual annotations, the dataset could be used for supervised pre-training or multi-task learning.

From an NLP methodology perspective, this data has the potential to investigate joint models that incorporate relationships between extracted aspects, related polar facts/opinions, surrounding context, and overall sentiment ratings. Existing pipelines make predictions using separate modules in isolation. Research along these lines may produce more explainable aspect-based sentiment analysis systems.

Our gold standard dataset also enables exploring the role of sentiments in collaborative knowledge-sharing forums. Recent studies have utilized sentiment analysis of Stack Overflow posts to investigate how emotions influence the success of questions [3], summarize developers' opinions about APIs [25], and provide relevant recommendations [17, 19]. However, these analyses were limited to overall sentiment. Our dataset's granular sentiment and aspect annotations open opportunities for more fine-grained modeling of how specific feelings toward distinct named entities' features impact developers' social forums.

6 SIMILAR DATASETS

There are many studies [5–8, 10, 12–14] on sentiment analysis and opinion mining but datasets are very scarce.

Two-dimensional Datasets. Only two datasets are available annotated with sentiment and aspects as the two dimensions.

Lin et al. [18] collected 1,662 sentences from Stack Overflow that were annotated by two independent coders manually to identify aspects and sentiments. Overall, the coders classified 523 sentences (31.5%) as containing at least one aspect. The remaining 1,130 sentences had no annotated aspects (and were considered neutral in sentiment). The annotated aspects and their frequencies (n) were as follows: community (n=10), compatibility (n=73), documentation (n=41), functional (n=246), performance (n=30), reliability (n=56), and usability (n=56). The dataset contains 373 and 150 sentences with positive and negative sentiments, respectively.

Uddin and Khomh [25] collected 4,522 sentences and identified the sentiments and aspects in those manually. They found 1,048 and 839 sentences were positive and negative in sentiments, respectively. The remaining 2,635 sentences were neutral. The annotated aspects and their frequencies (n) were as follows: performance (n=349), usability (n=1,438), security (n=164), bug (n=190), community (n=94), compatibility (n=94), documentation (n=254), legal (n=51), portability (n=71), and others (n=1,700).

One-dimensional Datasets. The following datasets are one-dimensional and annotated with either sentiments or emotions only. Noveilli et al. [21] collected 7,122 pull requests and commit comments from GitHub that were annotated by three human raters. According to the annotation, 28% of the comments were positive sentiment, 29% expressed negative sentiment, and the remaining 43% were labeled as neutral. Calefato et al. [2] collected 4,423 posts from Stack Overflow that were annotated with sentiments by three distinct human raters. In this dataset, 35% of posts conveyed positive sentiment, 27% expressed negative sentiment, and 38% of posts were neutral in sentiment. Lin et al. [19] obtained 1,500 sentences from Stack Overflow that were rated by two evaluators to identify sentiments in each sentence. In the dataset, 8.7% sentences were positive, 11.9% sentences expressed negative, and the remaining 79.4% sentences were neutral.

Ahmed et al. [1] manually annotated 2,000 code review comments to identify the sentiment polarities. However, the positive and neutral comments were merged into a single non-negative class in the publicly released version of the dataset. In the publicly available version of this dataset, 24.9% of the comments conveyed negative sentiment, and the remaining 75.1% expressed non-negative (neutral or positive) sentiment. Ortu et al. [23] collected 4,000 sentences from JIRA issue comments and rated those by three human raters with *love, joy, sadness, surprise*, and *anger.* The dataset contains 4.67% love, 2.95% joy, 8.02% sad, 0.7% surprise, and 8.5% anger comments. The remaining 75.23% comments were neutral.

In contrast to the above-mentioned datasets, our dataset, as released to the public [15], is the first, having four dimensions/tuples annotated with sentiments, polar facts, aspects, and named entities.

7 CONCLUSION

Sentiment analysis and opinion mining focused on software engineering texts can provide valuable insights into collaborative development barriers and enable enhanced analytics. However, progress has been constrained due to the lack of adequately large, multi-dimensional benchmark datasets adapted for this domain. Most ground truth datasets have focused solely on overall sentiment polarity or extracted aspects without interconnected facts and opinions. In this work, we have constructed and publicly released [15] the first four-dimensional gold standard dataset centered around the software engineering context. Through an opportunistic sampling approach and rigorous annotation process with six raters, we have developed a corpus of 2,000 Stack Overflow posts. These texts encompass sentiment labels across positive, negative, and neutral; fine-grained 15 aspects; related factual statements annotated separately as polar facts to avoid sentiment conflation; and named entity identification of five categories. The existing dimensions in the dataset support advancing opinion mining in software engineering.

REFERENCES

[1] T. Ahmed, A. Bosu, A. Iqbal, and S. Rahimi. 2017. Senticr: A Customized Sentiment Analysis Tool for Code Review Interactions. In *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*. IEEE, 106–111.

[2] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli. 2018. Sentiment Polarity Detection for Software Development. *Empirical Software Engineering* 23, 3 (2018), 1352–1382.

[3] F. Calefato, F. Lanubile, and N. Novielli. 2018. How to ask for technical help? Evidence-based guidelines for writing questions on Stack Overflow. *Information and Software Technology* 94 (2018), 186–207.

[4] J. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76, 5 (1971), 378–382.

[5] M. Islam, M. Ahmmed, and M. Zibran. 2019. MarValous: Machine Learning Based Detection of Emotions in the Valence-Arousal Space in Software Engineering Text. In *34th ACM/SIGAPP Symposium On Applied Computing (SAC)*. 1786–1793.

[6] M. Islam and M. Zibran. 2016. Exploration and Exploitation of Developers' Sentimental Variations in Software Engineering. *International Journal of Software Innovation* 4, 4 (2016), 35–55.

[7] M. Islam and M. Zibran. 2016. Towards Understanding and Exploiting Developers' Emotional Variations in Software Engineering. In *Proceedings of the International Conference on Software Engineering, Management, and Applications*. 185–192.

[8] M. Islam and M. Zibran. 2017. A Comparison of Dictionary Building Methods for Sentiment Analysis in Software Engineering Text. In *Proceedings of the International Symposium on Empirical Software Engineering and Measurement*. 478–479.

[9] M. Islam and M. Zibran. 2017. Leveraging Automated Sentiment Analysis in Software Engineering. In *Proceedings of the 14th International Conference on Mining Software Repositories*. 203–214.

[10] M. Islam and M. Zibran. 2018. A Comparison of Software Engineering Domain Specific Sentiment Analysis Tools. In *25th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 487–491.

[11] M. Islam and M. Zibran. 2018. Deva: Sensing Emotions in the Valence Arousal Space in Software Engineering Text. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. 1536–1543.

[12] M. Islam and M. Zibran. 2018. DEVA: Sensing Emotions in the Valence Arousal Space in Software Engineering Text. In *Proceedings of the ACM/SIGAPP Symposium On Applied Computing*. 1536–1543.

[13] M. Islam and M. Zibran. 2018. Sentiment Analysis of Software Bug Related Commit Messages. In *Proceedingd of the International Conference on Software Engineering and Data Engineering*. 3–8.

[14] M. Islam and M. Zibran. 2018. SentiStrength-SE: Exploiting Domain Specificity for Improved Sentiment Analysis in Software Engineering Text. *Journal of Systems and Software* 145 (2018), 125–146.

[15] Md Rakibul Islam, Md Fazle Rabbi, Jo Youngeun, Arifa Champa, Ethan Young, Camden Wilson, Gavin Scott, and Minhaz Zibran. 2024. New Opinion Mining Dataset. *https://doi.org/10.6084/m9.figshare.24779091* (2024).

[16] J. Klie, M. Bugert, B. Boullosa, R. Castilho, and I. Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, 5–9.

[17] B. Lin, N. Cassee, A. Serebrenik, G. Bavota, N. Novielli, and M. Lanza. 2022. Opinion Mining for Software Development: A Systematic Literature Review. *ACM Transactions on Software Engineering and Methodology*, 41 pages. https://doi.org/10.1145/3490388

[18] B. Lin, F. Zampetti, G. Bavota, M. Penta, and M. Lanza. 2019. Pattern-Based Mining of Opinions in QA Websites. In *Proceedings of the IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. 548–559. https://doi.org/10.1109/ICSE.2019.00066

[19] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto. 2018. Sentiment Analysis for Software Engineering: How Far Can We Go?. In *Proceedings of the 40th International Conference on Software Engineering*. 94–104.

[20] B. Liu. 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies* 5, 1 (May 2012), 1–167.

[21] N. Novielli, F. Calefato, D. Dongiovanni, D. Girardi, and F. Lanubile. 2020. Can We Use SE-specific Sentiment Analysis Tools in a Cross-Platform Setting?. In *Proceedings of the 17th International Conference on Mining Software Repositories*. ACM, 158–168.

[22] N. Novielli, D. Girardi, and F. Lanubile. 2018. A benchmark study on sentiment analysis for software engineering research. In *MSR '18: Proceedings of the 15th International Conference on Mining Software Repositories*. 364–375.

[23] M. Ortu, A. Murgia, G. Destefanis, P. Tourani, R. Tonelli, R. Tonelli, M. Marchesi, and B. Adams. 2016. The Emotional Side of Software Developers in JIRA. In *Proceedings of the 13th International Conference on Mining Software Repositories*. 480–483.

[24] M. Thelwall, K. Buckley, and G. Paltoglou. 2012. Sentiment Strength Detection for the Social Web. *J. Am. Soc. Inf. Sci. Technol.* 63, 1 (2012), 163–173.

[25] G. Uddin and F. Khomh. 2019. Automatic mining of opinions expressed about APIs in Stack Overflow. *IEEE Transactions on Software Engineering* (2019), 522–559.

[26] D. Ye, Z. Xing, C. Foo, Z. Ang, J. Li, and N. Kapre. 2016. Software-Specific Named Entity Recognition in Software Engineering Social Content. In *Proceedings of the IEEE International Conference on Software Analysis, Evolution and Reengineering*. 90–101.