An Emoji-Aware Analysis of Human Sentiment in Agentic Code Review

Karina Kohl
Institute of Informatics
UFRGS
Porto Alegre, RS, Brazil
karina.kohl@inf.ufrgs.br

Anielle Andrade
Lab. of Empirical Studies in SE (LESSE)
UNIPAMPA
Alegrete, RS, Brazil
anielleandrade@unipampa.edu.br

Abstract

Automated agents now author a growing share of pull requests in open-source repositories, yet human reviewers remain responsible for evaluating these contributions. Little is known about how developers express sentiment when reviewing agent-generated code. We analyze review comments from AIDev, a large-scale MSR Mining Challenge dataset capturing agent-authored pull requests and review interactions across real-world GitHub repositories, using an emoji-aware approach that combines VADER scores with a custom emoji lexicon. Human feedback on both agent- and human-generated pull requests is predominantly positive or neutral. Emojis appear mainly as secondary signals that reinforce approval, while automated reviewers rarely use emojis and communicate almost entirely through structured text. Sentiment trajectories differ slightly by context: human reviewers show a mild positive slope when reviewing agent-generated pull requests, whereas automated reviewers show a slight decline. Effect sizes are small throughout. These results suggest that developers engage with agent-generated contributions in a broadly constructive way, though communication norms differ between human and automated reviewers.

Keywords

agentic systems, code review, sentiment analysis, software repositories, mining software repositories

1 Introduction

Advances in artificial intelligence (AI) and large language models have enabled the emergence of *agentic* software development tools that can autonomously generate code changes and submit pull requests (PR)—formal proposals to merge proposed code modifications into a version-controlled repository. As a result, modern software repositories are increasingly populated with contributions authored by automated agents rather than by human developers. Despite this shift, human reviewers remain central to the code review process, as they are ultimately responsible for evaluating correctness, maintainability, and suitability before changes are integrated into the codebase.

Code review is not only a technical quality assurance activity but also a fundamentally social process. Prior research shows that emotions and sentiment expressed during collaborative software development influence productivity, coordination, task quality, creativity, and job satisfaction [8, 9]. Prior work argues that awareness of emotional signals can help developers and managers better understand team dynamics [9] and take corrective actions. In this context, sentiment expressed during code review can signal trust, acceptance, or concern toward a proposed change.

The growing presence of agent-generated pull requests raises new questions about human–AI collaboration in software engineering. While prior work has examined how AI-powered tools reshape developers' roles and shift effort toward reviewing machine-generated suggestions [6], little is known about how developers emotionally respond to code authored by automated agents. These affective responses matter because trust, accountability, and the quality of reviews are directly at stake when automated agents author code. For example, do reviewers react differently to agent-generated versus human-generated code? Does the emotional tone of review discussions evolve differently depending on the origin of the contribution?

In this paper, we investigate how human developers and automated agents express sentiment during the review of both agent-generated and human-generated pull requests. We analyze sentiment expressed in review comments and examine how emotional signals differ across reviewer types and pull request origins. To capture both affective cues commonly present in code review discussions, we adopt an emoji-aware sentiment analysis approach that combines textual sentiment with emoji-based signals.

In addition to large-scale quantitative analysis, we include illustrative qualitative examples to show how these sentiment patterns appear in practice. We address the following research questions:

- RQ1. *What is the sentiment of human and agent review comments on agent-generated pull requests?*
- RQ2. *What is the sentiment of human and agent review comments on human-generated pull requests?*
- RQ3. *What are the sentiment trajectories in Code Review?*
- RQ4. *What do these sentiment patterns look like in practice?*

Our contributions are threefold: (1) a large-scale empirical analysis of developer sentiment and dynamic discussion trajectories across both agent- and human-generated pull requests in popular GitHub repositories; (2) an emoji-aware sentiment analysis approach that integrates lexical polarity with domain-informed emoji weightings to capture secondary affective signals in technical discourse; and (3) a purposively sampled qualitative investigation illustrating how these sentiment patterns and communication styles—ranging from lightweight conversational approvals to structured diagnostic reports—manifest in real-world code review interactions.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the dataset. Section 4 presents the data analysis approach. Section 5 reports results for each research question. Section 6 discusses the findings and addresses threats to validity. Section 7 address limitations and, Section 8 concludes.

2 Background and Related Work

We structure our analysis around three core dimensions: the social and collaborative dynamics embedded within peer code review practices, the communicative function of non-verbal signals and implicit cues in technical discourse, and the transformative integration of automated agents into the modern verification workflow.

2.1 Social Dynamics in Code Review

Code review is a quality assurance practice in which developers examine changes before integration [2, 5]. Beyond technical verification, this practice serves as a social interaction in which developers express their sentiments directly. Consequently, participants' affective state influences engineering outcomes [15]. Prior research links emotions expressed during collaborative work to productivity and task quality [5]. Specifically, empirical evidence indicates that negative affective patterns directly prolong review intervals and decrease the likelihood of code acceptance [3]. Furthermore, toxic interactions reduce productivity and increase the risk of developer burnout [18]. Demographic factors also influence these dynamics; for example, female developers are statistically less likely to express strong sentiments than their male counterparts [15].

To quantify affective expressions in software repositories, researchers employ sentiment analysis, a technique that assigns polarity to textual artifacts [9] to evaluate developer attitudes toward technical entities [2, 12]. However, relying exclusively on conventional lexical parsing limits our understanding of modern review dynamics, as developers frequently incorporate non-textual cues into their technical communication [8, 9]. To capture these broader dynamics, we adopt VADER [10] over domain-specific tools such as SentiCR [2]. While SentiCR is tailored for technical code review discourse, it primarily outputs categorical sentiment classifications; in contrast, VADER generates a continuous compound score ranging from -1.0 to $+1.0$. This continuous scale is mathematically essential for integrating textual polarity with emoji-based signals under the weighted composite scoring scheme defined in Section 4.

In asynchronous software engineering discourse, written feedback lacks the prosodic cues of speech—such as vocal tone and volume—which significantly increases the risk of critical suggestions being perceived as harsh or confrontational [18]. To preemptively reduce affective ambiguity and soften the emotional impact of technical critique, developers routinely integrate lightweight paralinguistic cues into their review comments [15].

2.2 Non-Verbal Signals in Code Review

Recent studies extend software engineering sentiment analysis beyond plain text by investigating the role of emojis in collaborative workflows. In pull requests and code reviews, developers frequently utilize emoji reactions as non-verbal, paralinguistic cues that supplement written communication [1, 21]. While these visual artifacts convey critical affective feedback, empirical evidence confirms that they complement rather than replace substantive textual discussion [21]. Despite their prevalence, however, prior work has rarely examined the dynamic interplay between explicit emoji usage and underlying lexical polarity. In particular, understanding how these non-verbal cues modulate the perceived tone of technical critiques remains an open methodological and empirical challenge.

Specifically, emojis often act as 'mitigation markers.' Research indicates that reviewers employ positive emojis (e.g., 'thumbs up' or 'smile') to soften critical comments or reinforce approval, maintaining social cohesion without compromising technical rigor [21]. Disregarding these nonverbal signals leads to an incomplete analysis of review sentiment. Therefore, our emoji-aware approach integrates both the lexical content and the modifying effect of accompanying emojis to ensure accurate detection.

Context determines the function of these symbols. While developers use emojis in documentation to emphasize technical content, their application in pull requests primarily communicates sentiment [13]. These signals capture specific emotional states, such as joy or anger, rather than simple binary polarity [7]. To support this communication, platforms introduced formal 'reaction' mechanisms. Developers use these features to provide low-noise feedback, separating emotional expression from technical discussion [4]. Effective analysis must therefore distinguish between these functional and affective roles. Yet, while these patterns define human-to-human collaboration, the extent to which they persist in interactions with non-human agents remains an open question.

2.3 Human Interaction with Automated Agents

In this study, we use the term *agentic* to characterize software systems that transition beyond passive code recommendation to autonomously plan and execute complex, multi-step engineering tasks. Within collaborative repository workflows, these systems independently generate code modifications and submit formal pull requests without continuous human intervention [20].

The integration of artificial intelligence alters code review practices. Research indicates that AI-powered pair programming shifts developer effort from code authoring to evaluating machine-generated suggestions, often increasing the time allocated to review [6, 20]. This shift raises questions about monitoring AI-generated code. Tracking agent-authored contributions is necessary to understand the effects on build failures, test outcomes, and post-release defects, and to calibrate review scrutiny [6, 14, 16]. Beyond technical metrics, these concerns prompt an investigation into the social norms developers apply when responding to non-human contributors.

Code review automation relied on rule-based bots to flag style violations or coverage gaps. Developers typically treated these tools as functional instruments rather than social counterparts [23]. However, recent advancements in deep learning have shifted automation toward generative tasks designed to imitate human developers, such as synthesizing natural-language review comments and proposing complex code modifications [19]. This evolution challenges established social boundaries. While developers maintain polite norms with human peers to preserve professional relationships, it remains unclear if this social contract extends to machines that emulate human behavior. Consequently, this study investigates whether the contributor's identity, human or agent, alters the socio-affective dynamics of the review. Specifically, we analyze textual sentiment and the use of nonverbal mitigating signals (e.g., emojis) to determine whether developers apply the same politeness norms to AI as they do to human peers.

3 Dataset

We base our study on AIDev [11], a large-scale dataset of agent-authored pull requests mined from real-world GitHub repositories and released as part of the MSR Mining Challenge. AIDev contains 932,791 Agentic-PRs spanning 116,211 repositories and 72,189 developers, covering contributions generated by five AI agents: Claude Code, Cursor, Devin, GitHub Copilot, and OpenAI Codex.

For this paper, we analyze a curated subset of AIDev, referred to as *AIDev-pop*, which includes only repositories with more than 100 GitHub stars. This filtering reduces the dataset to 33,596 Agentic-PRs from 2,807 popular repositories and ensures consistent availability of review and comment data, which is essential for studying review dynamics. Within AIDev-pop, we merged entries from the `pr_reviews` (28,875 entries) and `pr_review_comments` (26,868 entries) tables into a unified review interaction dataset, yielding 37,235 review-related entries used in our analyses.

To enable comparative analyses between agent- and human-authored contributions, we also required review comments from human-authored pull requests. Because the base dataset provides metadata for human-authored PRs without their corresponding review discussions, we developed a Python script to retrieve all associated review comments and review states using the GitHub GraphQL API.

We also used a large language model as an auxiliary tool to suggest candidate automation accounts based on common naming patterns; all suggestions were validated manually. The final classification of reviewers as human or automated was validated manually by the authors. We identified automated reviewers using a rule-based approach over GitHub usernames. Specifically, we flagged accounts whose login matched (i) explicit bot markers (e.g., suffix "[bot]" or the token "bot"), (ii) automation-related keywords (e.g., "actions", "agent"), and (iii) a curated list of known automation and AI-reviewer accounts observed in GitHub pull request workflows. The script records which rule triggered each classification to support transparency and reproducibility.

4 Data Analysis

For RQ1, we analyzed the affective tone of pull request feedback using a two-stage sentiment analysis approach. First, we applied VADER (Valence Aware Dictionary and sEntiment Reasoner) [10], a lexicon- and rule-based sentiment analysis tool, to all review-related textual content, separately for reviews authored by human developers and by automated agents. VADER maps lexical features to sentiment intensity scores and calculates a normalized compound score ranging from -1.0 (extremely negative) to +1.0 (extremely positive). We selected VADER due to its established effectiveness on short, informal, and opinionated text, which closely resembles the linguistic style of code review discussions. This step provides a text-only baseline sentiment signal for both human- and agent-generated feedback.

In a second step, we extended VADER with a custom emoji lexicon, motivated by the use of emojis as affective cues in developer communication. Grounded in prior empirical work showing that emojis in pull requests serve primarily as secondary affective amplifiers rather than primary carriers of meaning [**? ?**], we computed a composite sentiment score that prioritizes textual content while still incorporating emoji-based affect. Emojis commonly observed in review discussions were mapped to domain-informed sentiment values (e.g., 👍 → +2, 👎 → −2, 🚀 → +3, 😬 → −1), while ambiguous emojis (e.g., 🤖) were treated as neutral. The final score is defined as:

$$FinalScore = (0.7 \times VADER) + (0.3 \times EmojiScore).$$

This weighting assigns greater influence to textual sentiment while allowing emojis to modulate the overall score without over-amplifying isolated emoji usage. For example, the comment *"I'm not sure the agent considered the edge cases* 😬*"* receives a near-neutral VADER score (−0.02), but the negative emoji shifts the composite score to −0.47, providing face validity by more accurately capturing reviewer concern. Conversely, *"LGTM* 🚀*"* combines a moderately positive textual sentiment with a strongly positive emoji, yielding a clearly positive final score.

Once automated agents rarely use emojis in their review feedback (as shown in Tables 1 and 2, where *emoji_mean* is nearly zero for automated reviewers), the composite score for agent-generated comments is functionally equivalent to the VADER-only baseline. Thus, our central comparative analysis between human and agent reviewers remains robust and is unlikely to be affected by alternative weighting schemes. The same scoring approach applies to RQ2, where we analyze comments on human-generated pull requests.

To address RQ3, we analyzed how sentiment evolves throughout the review lifecycle of a pull request. For each pull request, we ordered all review-related comments by their submission timestamp. Using the composite sentiment score described above, we then computed three trajectory-based metrics: (i) the sentiment of the first review comment, (ii) the sentiment of the last review comment, and (iii) the sentiment slope, defined as the linear trend of sentiment across successive comments within the pull request. The sentiment slope captures whether reviewer sentiment becomes more positive or more negative as the review progresses.

We compared sentiment slopes between human-generated and agent-generated pull requests using the Mann-Whitney U test, a non-parametric alternative suitable for skewed distributions. To quantify the magnitude of observed differences, we report effect sizes using Cliff's delta (δ).

For RQ4, we conducted a qualitative analysis that is descriptive and illustrative rather than inductive. Our objective was not to derive emergent themes or construct theory, but to contextualize and exemplify the quantitative sentiment findings by inspecting representative developer comments. Accordingly, we do not perform a formal thematic analysis nor claim theoretical saturation; instead, we present selected examples that illustrate how sentiment polarity and emoji usage manifest in daily code review interactions.

To select these examples, we employed purposive sampling stratified across all four reviewer-type × pull request origin combinations. Within each of the four experimental cohorts, we selected representative comments spanning the three primary sentiment categories observed in the quantitative data: positive, neutral, and negative. To ensure our qualitative illustrations reflect general repository behavior rather than anomalous artifacts, we selected comments typical of the most frequent textual and emoji patterns rather than extreme outliers. The initial selection was conducted by one author and independently cross-checked by a second author.

5 Results

5.1 RQ1. Sentiment on Agent-Generated Pull Requests

Human feedback on agent-generated pull requests is predominantly positive or neutral, indicating a constructively oriented review culture. As shown in Table 1, positive feedback constitutes the largest proportion of human review comments, achieving a composite mean score of 0.362. This positive affect is driven primarily by lexical textual sentiment (*VADER mean* = 0.462), while emojis provide a secondary reinforcement effect (*emoji mean* = 0.130). Meanwhile, neutral comments cluster tightly around zero, characteristic of objective, informational, or diagnostic code discussions. Conversely, negative comments appear significantly less frequently and exhibit moderate affective polarity (composite mean = −0.287), which is again conveyed almost entirely through structured textual critique. Overall, emoji usage within human feedback remains modest and structurally asymmetric—acting primarily as an amplifier for approval and encouragement rather than as a vehicle for technical criticism or disapproval.

Agent feedback on agent-generated pull requests follows a similarly constructive overarching distribution, but exhibits significantly stronger lexical polarity. Automated review comments classified as positive demonstrate markedly higher textual positivity (*VADER mean* = 0.602) and a correspondingly higher composite sentiment score (0.427) than equivalent human feedback (0.362). However, this elevated score is driven entirely by written vocabulary; emojis contribute a negligible signal to agent feedback (*emoji mean* = 0.019, compared to 0.130 for humans). This near-complete absence of visual paralinguistic cues indicates that automated reviewers rely almost exclusively on formal, structured textual phrasing rather than informal affective markers to communicate approval and verification.

For negative comments, agents also exhibit stronger textual negativity (VADER mean = −0.566) than humans (−0.409), resulting in a more negative composite score (−0.395 vs. −0.287). Neutral comments from both sources remain centered around zero, with minimal variance, indicating that both human and automated reviewers frequently provide informational or descriptive feedback with minimal emotional tone.

We compared the scores of human and agent review comments on agent-generated pull requests using the non-parametric Mann–Whitney U test. To quantify the magnitude of these differences on a compatible, rank-based scale, we reported the effect size using Cliff's delta (δ), interpreted according to the thresholds defined by Romano et al. [17]: negligible ($|\delta| < 0.147$), small ($0.147 \leq |\delta| < 0.33$), medium ($0.33 \leq |\delta| < 0.474$), and large ($|\delta| \geq 0.474$) (Table 1).

All differences were statistically significant ($p < .001$), but effect sizes were small: Cliff's delta indicates a small effect for positive comments ($\delta = -0.183$), negligible for neutral comments ($\delta = 0.029$), and small-to-moderate for negative comments ($\delta = 0.303$). Thus, while sentiment distributions differ significantly, practical differences are modest and concentrated mainly in negative interactions. Overall, human reviewers generally engage with agent-generated pull requests in a constructive and restrained manner, relying primarily on textual cues to express affect, with emojis acting as a secondary signal that mainly reinforces positive sentiment.

Summary for RQ1: Human comments on agent-generated pull requests are predominantly positive or neutral, reflecting constructive human–AI collaboration. While sharing this overall distribution, automated agents display stronger textual polarity—being more intense in both praise and critique—and communicate exclusively through text without the supportive emojis occasionally used by human reviewers.

5.2 RQ2. Sentiment on Human-Generated Pull Requests

Human feedback on human-generated pull requests is also predominantly positive or neutral. As shown in Table 2, positive comments form the largest category (composite mean = 0.339), driven mainly by textual sentiment (VADER mean = 0.462) and reinforced by emojis (emoji mean = 0.534). Neutral comments center tightly around zero (composite mean ≈ 0), reflecting routine informational and technical exchanges. Negative comments are less frequent and moderately negative (composite mean = −0.260), again expressed primarily through text.

Agent feedback on human-generated pull requests follows a similar distribution but exhibits stronger textual polarity. Agent comments classified as positive show higher textual positivity (VADER mean = 0.590) and a higher composite sentiment score (0.415) compared to human comments (0.339). However, emojis are almost absent in agent feedback (emoji mean = 0.0068), suggesting that automated reviewers rely almost exclusively on textual sentiment cues.

For negative comments, agents again display stronger textual negativity (VADER mean = −0.541) than humans (−0.365), resulting in a more negative composite score (−0.379 vs. −0.260). Neutral comments from both sources remain centered around zero with minimal variation, indicating that both human and automated reviewers frequently provide factual or procedural feedback without a strong emotional tone.

We compared human and agent composite sentiment scores on human-generated pull requests using the Mann–Whitney U test and Cliff's delta (Table 2). Differences were statistically significant for negative and positive comments ($p < .001$), but not for neutral comments ($p = 0.096$). Effect sizes were medium for negative ($\delta = 0.391$), small for positive ($\delta = -0.243$), and negligible for neutral comments ($\delta = 0.021$). These results confirm that agent feedback is more strongly polarized in both directions, while neutral interactions remain comparable.

Summary for RQ2: Comments on human-generated pull requests are predominantly positive or neutral, demonstrating broadly comparable overarching sentiment dynamics across reviewer types. However, automated agents exhibit stronger textual polarity in both praise and critique, and rely exclusively on text without the supportive emojis used by humans.

Table 1: Sentiment Analysis of Human and Agent Feedback on Agent-generated Pull Request. Mann–Whitney U tests and Cliff's δ compare Human and Agent comments within each sentiment label using the composite score.

source	sentiment	count	vader_mean	emoji_mean	composite_mean	*pvalue*	Cliff's δ
Human	negative	2468	-0.409	-0.003	-0.287		
Agent	negative	3235	-0.566	0.006	-0.395	$p < .0000$	0.303 (small)
Human	neutral	7683	-0.000	0.000	-0.000		
Agent	neutral	1871	-0.002	0.003	-0.001	$p = .0000$	0.029 (negligible)
Human	positive	6448	0.462	0.130	0.362		
Agent	positive	10438	0.602	0.019	0.427	$p < .0000$	-0.183 (small)

Table 2: Sentiment Analysis of Human and Agent Feedback on Human-generated Pull Request. Mann–Whitney U tests and Cliff's δ compare Human and Agent comments within each sentiment label using the composite score.

source	sentiment	count	vader_mean	emoji_mean	composite_mean	*pvalue*	Cliff's δ
Human	negative	907	-0.365	-0.006	-0.260		
Agent	negative	656	-0.541	0.000	-0.379	$p < .0000$	0.391 (medium)
Human	neutral	2595	0.0004	-0.0013	-0.00016		
Agent	neutral	340	0.00065	0.000	0.00046	$p = 0.096$	0.021 (negligible)
Human	positive	2261	0.462	0.534	0.339		
Agent	positive	1029	0.590	0.0068	0.415	$p < .0000$	-0.243 (small)

5.3 RQ3: Sentiment Trajectories in Code Review

5.3.1 ***Trajectories on Agent-generated pull request.*** Table 3 summarizes sentiment trajectory characteristics for reviews of agent-generated pull requests. On average, pull requests reviewed by humans and agents received comparable numbers of comments (5.9 and 5.4, respectively). However, the evolution of sentiment across the review discussion differs slightly between the two sources.

Human review comments show a small positive change in sentiment between the first and last comment ($\Delta = 0.037$) and a positive average slope (0.018), indicating that discussions involving human reviewers tend to become slightly more positive as the review progresses. In contrast, agent review comments exhibit a small negative sentiment change ($\Delta = -0.022$) and a negative slope (-0.012), suggesting a mild decline in sentiment throughout the review interaction.

A Mann–Whitney U test indicates that sentiment slopes differ significantly between human and agent comments ($p < 0.001$), but with small effect ($\delta = 0.09$) showing this practical difference is limited. Overall, review sentiment on agent-generated pull requests becomes slightly more positive over time with human participation, whereas automated reviewer interactions show a modest downward trend.

Summary for RQ3.1: Review discussions on agent-generated pull requests evolve differently by reviewer type: human sentiment gradually improves over time, whereas automated reviewers exhibit a mild decline. However, a small statistical effect size suggests these overarching trajectories remain broadly comparable.

5.3.2 ***Sentiment Trajectories in Human-generated Pull Requests.*** Table 4 summarizes sentiment trajectory characteristics for reviews of human-generated pull requests. On average, pull requests reviewed by humans receive slightly more comments (7.68) than those reviewed by agents (6.63), suggesting somewhat longer review discussions among human participants.

Both human and agent reviewers exhibit positive sentiment trajectories throughout the review process. Human reviewers show a slightly higher mean sentiment change between the first and last comment ($\Delta = 0.035$) and a marginally higher average sentiment slope (0.008), indicating that discussions involving human reviewers tend to become somewhat more positive as the review progresses. Agent reviewers also display a positive trajectory ($\Delta = 0.010$, slope = 0.006), though the magnitude of this increase is smaller.

A Mann–Whitney U test comparing the distribution of sentiment slopes between human and agent reviewers does not reveal a statistically significant difference ($p = 0.404$). The associated effect size, measured using Cliff's delta, is negligible ($\delta = 0.034$). These results suggest that the evolution of sentiment during the review of human-generated pull requests is broadly similar for human and automated reviewers.

Summary for RQ3.2: Both human and agent reviewers exhibit slightly improving sentiment trajectories when reviewing human-generated pull requests. Although humans show marginally higher positive sentiment gain, this difference is not statistically significant and the effect size is negligible, indicating largely comparable emotional evolution across reviewer types.

Table 3: Summary of sentiment trajectory metrics agent-generated pull requests.

	mean comments	mean Δ	mean slope	trend
Human	5.9	0.037	0.018	↑
Agent	5.4	-0.022	-0.012	↓

Notes. $U = 3{,}440{,}392$, $p < 0.001$, $\delta = 0.09$

Table 4: Summary of sentiment trajectory metrics for human-generated pull requests.

	mean comments	mean Δ	mean slope	trend
Agent	6.627	0.010	0.006	↑
Human	7.682	0.035	0.008	↑

Notes. $U = 103{,}176.5$, $p = 0.404$, $\delta = 0.034$

5.4 RQ4. Sentiment Patterns in Practice

To contextualize the quantitative findings, we examined representative review comments on agent- and human-generated pull requests, focusing on how sentiment and emoji usage are manifested in practice.

5.4.1 ***Human review comments on Agent-generated pull requests.*** Human review comments on agent-generated pull requests are typically short, conversational, and socially expressive. Positive feedback often appears as brief approval statements or emoji reactions that signal agreement with minimal textual explanation. For example:

> *"Wow!! LGTM* 👍*"*

These comments illustrate how human reviewers frequently use emojis as lightweight approval signals that reinforce positive sentiment. In many cases, emojis complement short textual responses, allowing reviewers to acknowledge acceptable changes quickly without extensive explanation.

Neutral comments often appear as brief acknowledgments or reactions that indicate engagement but convey little emotional polarity. For instance:

> "⭐"

Such reactions function primarily as social signals rather than substantive feedback.

Negative feedback from human reviewers tends to focus on specific technical issues rather than emotional expression. For example:

> *"It failed: https://github.com/..."*

> *"No issues with the way this is implemented, but ..."*

Even when criticism is present, the tone remains task-focused and professional, reflecting established norms of constructive code review.

5.4.2 ***Agent review comments on Agent-generated pull requests.*** Agent-generated review comments on agent-generated pull requests differ in style and structure. Rather than short conversational responses, agent comments tend to be longer, structured, and report-like, often providing summaries of the pull request or automated analyses.

For example, some agent comments begin with structured summaries:

> *"Greptile Summary This PR introduces a comprehensive ..."*

or

> *"Pull Request Overview This PR improves logging functionality ..."*

In other cases, agents generate structured reports that summarize detected issues or suggested improvements:

> *"Actionable comments posted: 11"*

Agent comments may also include structured bug reports describing potential problems in the code:

> *"Bug: Incorrect Implementation ..."*

These examples illustrate how agent-generated comments resemble automated analysis reports rather than conversational feedback. Unlike human reviewers, agents rarely use emojis and instead communicate sentiment almost entirely through structured textual explanations.

Overall, while both humans and agents participate in reviewing agent-generated pull requests, their communication styles differ. Human reviewers tend to combine technical feedback with lightweight social signals, such as emojis, whereas automated reviewers produce longer, more structured comments that summarize changes and identify issues.

5.4.3 ***Human review comments on Human-generated pull requests.*** Human review comments on human-generated pull requests tend to be conversational, collaborative, and technically focused, often mixing short acknowledgements with specific implementation suggestions. Positive feedback often takes the form of expressions of appreciation or confirmation that changes have been implemented.

For example, some comments express gratitude after updates:

> *"Thank you.* 🙏 *I've removed the env section."*

Human reviewers also commonly provide direct improvement suggestions, phrased as conversational requests:

> *"Can you wrap this in a try/catch, after that you can rethrow the error?"*

Other comments include short technical clarifications or explanations about implementation details:

> *"Because prisma can have a related model as a field..."*

These illustrative examples demonstrate that human reviewers prioritize collaborative problem-solving, blending concise technical guidance with paralinguistic social signals, such as emojis. Even when identifying flaws or requesting substantive revisions, human feedback maintains a distinctly constructive tone, centering on shared project goals and collective improvement of the implementation. By balancing professional critique with these informal affective cues, human reviewers reinforce team rapport and mitigate the potential for friction often inherent in asynchronous technical review.

5.4.4 ***Agent review comments on Human-generated pull requests.*** Agent-generated review comments on human-generated pull requests exhibit a more structured and diagnostic style. Rather than conversational feedback, agent comments often resemble automated analysis reports that highlight potential issues, refactoring opportunities, or stylistic improvements.

For instance, many agent comments explicitly flag potential problems:

> "⚠️ *Potential issue with the time threshold logic…*"

Other comments present structured suggestions for improving the code:

> "🛠️ *Refactor suggestion: address package structure…*"

Agent comments may also include labeled sections that describe the type of feedback being provided:

> "🧹 *Nitpick (assertive): device logout changes…*"

Unlike human reviewers, agents rarely engage in conversational exchanges or acknowledgements. Instead, their comments focus on structured diagnostics and improvement suggestions, often formatted with headings or categories that clarify the nature of the detected issue.

Overall, while both humans and agents contribute actionable feedback when reviewing human-generated pull requests, their communication styles differ. Human reviewers emphasize collaborative dialogue and contextual explanations, whereas automated reviewers tend to produce structured reports that prioritize clarity, consistency, and issue detection.

Summary for RQ4: Across all scenarios, review discussions remain predominantly constructive and neutral-to-positive. While human reviewers use a conversational style with emojis to reinforce approval, agents produce longer, structured comments resembling automated diagnostic reports. Thus, while the emotional tone remains similar, humans rely on social cues and conversational language, whereas agents communicate sentiment primarily through structured text.

6 Discussion

Prior sentiment studies in code review [3, 9, 15] focus exclusively on human-to-human interactions. To our knowledge, we are the first to characterize developer sentiment specifically in the context of agent-generated pull requests at scale, using a real-world dataset (AIDev). Moving from human-only teams to AI-integrated workflows raises important socio-technical questions about how affective norms evolve when code is authored by machines.

In broader human–computer interaction, research shows that people prefer human empathy over AI empathy—even when they rate AI empathy higher [22], suggesting that social norms do not align with a simple preference for technical competence. To understand whether similar interpersonal dynamics hold in code review, where developers must evaluate contributions from automated agents, we examined sentiment expressions across both review contexts. Two consistent patterns emerge from this analysis. First, regarding overall tone, review comments—whether authored by humans or agents—are predominantly positive or neutral, indicating that code review interactions remain largely constructive regardless of contribution origin. Second, regarding emotional intensity, agent-generated feedback exhibits significantly stronger textual polarity than human feedback: agents produce more strongly positive comments when expressing approval and more strongly negative comments when identifying issues. This polarity gap appears across both contexts but is especially pronounced when agents review human-generated pull requests, where the effect size for negative comments reaches a medium magnitude ($\delta = 0.391$), compared to the smaller effects observed on agent-generated code.

These findings align with prior work suggesting that code review discourse is typically task-focused and normatively polite, even when disagreements arise [9]. Overall, these results suggest that developers interact with AI-generated contributions in ways that resemble traditional human–human code review dynamics, maintaining constructive and professional communication patterns.

Another consistent difference concerns the use of emojis. Human reviewers use emojis to reinforce positive feedback while softening critical remarks, whereas agent comments almost never include emojis, relying almost entirely on textual cues to convey affect. The incorporation of emoji-aware sentiment analysis, therefore, provides an important complementary signal for understanding review interactions. Emojis appear to function primarily as affective amplifiers: they reinforce positive feedback (e.g., approval or encouragement) and occasionally soften critical remarks, while negative feedback itself is conveyed largely through textual and technical language. This asymmetric use of emojis mirrors findings in [21], which show that emoji reactions in pull requests complement rather than replace substantive discussion.

The emoji asymmetry raises a practical question about how automated reviewers communicate criticism. Prior work shows that emojis soften negative feedback and reinforce approval in pull request discussions [21]. When an agent flags an issue without any mitigating signal, the comment may read as more blunt than an equivalent human comment, even if the textual content is similar in polarity. Whether this difference in style affects how developers receive and act on agent feedback is not clear from our data; it warrants direct investigation through controlled experiments or developer interviews.

Despite these stylistic differences, the overall sentiment structure of review interactions remains broadly similar across scenarios, with neutral comments centered around zero and statistical differences generally accompanied by small or moderate effect sizes. Together, these results suggest that while automated reviewers display slightly stronger sentiment polarity and a more text-centric communication style, they participate in code review exchanges in ways that remain largely aligned with human review dynamics. In the context of agent-generated pull requests, emojis thus appear to function as lightweight social signals that help maintain a collegial tone rather than as primary carriers of criticism or concern.

The trajectory analyses reveal asymmetry in how sentiment evolves during review discussions. While El Asri et al. [3] link sentiment to review outcomes, no prior work analyzes how sentiment evolves within individual review discussions across reviewer types and PR origins. In our data, when reviewing agent-generated pull

requests, human reviewers exhibit slightly improved sentiment trajectories, whereas agent reviewers show a mild decline in sentiment over the course of the discussion. Although this difference is statistically significant, the effect size is small, indicating only modest practical differences. In contrast, when reviewing human-generated pull requests, both human and agent reviewers exhibit similarly positive sentiment trajectories, with no statistically significant difference observed.

That said, a broadly positive tone does not tell us whether developers are indifferent to the origin of a contribution or simply maintain professional norms regardless. Research on human–AI interaction suggests that people sometimes prefer human-sourced feedback even when they rate AI-generated responses as equivalent or superior [22]. If similar preferences operate in code review, the positive tone we observe may reflect politeness conventions rather than genuine acceptance of agent-generated code. Sentiment analysis alone cannot distinguish between these interpretations; survey-based or interview methods would be needed to separate them.

This pattern suggests that sentiment dynamics during code review are largely stable across reviewer types when interacting with human-generated contributions, but diverge slightly when automated reviewers interact with agent-generated code. One possible interpretation is that automated reviewers adopt a somewhat stricter or more corrective tone when evaluating agent-generated contributions, whereas discussions involving human-authored code maintain a consistently constructive trajectory regardless of reviewer type. Overall, these results indicate that while automated reviewers participate in review conversations with sentiment dynamics broadly similar to humans, subtle differences emerge in interactions involving automated code generation.

A further concern is whether VADER, which was designed for social media text [10], fits the register of code review comments. Review comments are often terse, imperative, or domain-specific, which may cause VADER to classify substantive feedback as neutral when it carries technical weight. The high share of neutral comments in our data (e.g., 7,683 of 16,599 human comments on agent-generated pull requests) may partly reflect this mismatch rather than true affective neutrality. Our emoji extension mitigates the problem for comments that include emojis, but most comments do not. Whether tools calibrated for technical discourse, such as SentiCR [2], produce systematically different classifications on this data remains an open question.

Consistent qualitative patterns emerge regarding how humans and agents express sentiment during code review interactions. First, human reviewers tend to communicate in a conversational and collaborative style, often combining short technical suggestions with lightweight social signals such as emojis (e.g., 👍, ⭐, 🙏). These reactions frequently accompany brief approval statements such as "LGTM", serving as quick acknowledgements that reinforce positive sentiment without requiring lengthy explanations.

Second, agent-generated review comments follow a markedly different communication style. Rather than conversational responses, agent comments typically appear as structured diagnostic reports or summaries of the pull request. These comments often include labeled sections, categorized suggestions, or automatically generated issue descriptions. As a result, agent feedback tends to be longer and more structured, focusing on technical analysis rather than social interaction.

Third, the interaction dynamics differ slightly depending on the origin of the pull request. When reviewing agent-generated pull requests, human reviewers frequently use brief approvals and emojis to acknowledge acceptable automated changes, suggesting that these reviews are often lightweight confirmations rather than extended discussions. In contrast, agent reviewers tend to generate structured summaries and automated issue reports when reviewing agent-generated contributions.

Finally, when reviewing human-generated pull requests, both human and agent reviewers provide more detailed technical feedback. Human reviewers typically frame suggestions in a conversational tone that reflects collaborative problem-solving, whereas agent reviewers continue to produce structured diagnostic comments that highlight potential improvements or issues.

Taken together, these qualitative observations reinforce our quantitative results: while the overall emotional tone of review discussions remains resiliently constructive across all scenarios, humans and agents express sentiment through distinctly different communication styles. Human reviewers rely on conversational language and social signals to convey affect, whereas automated reviewers communicate sentiment primarily through structured textual analyses.

Synthesizing these socio-technical dynamics reveals exactly how our work converges with, and departs from, the existing literature. Our findings converge with prior research in two key respects: code review discourse remains predominantly constructive regardless of PR origin, consistent with El Asri et al. [3], and emojis serve to complement rather than replace textual discussion, consistent with Wang et al. [21]. However, our empirical analysis breaks new ground by identifying two phenomena with no prior precedent in the literature: the emoji asymmetry between human and automated reviewers, and the diverging sentiment trajectories that emerge when automated reviewers interact with agent-generated code.

Figure 1 illustrates the summary of the entire process of the study, since methodology to reach the results. The NotebookLM tool was used to generate the image.

6.1 Threats to Validity

We structure our discussion of validity following Wohlin et al. [24].

Construct validity may be affected by the specific tools and heuristics used for classification. First, because VADER is optimized for social media, it might incorrectly categorize terse, imperative, or highly technical code review feedback as neutral. Second, our composite sentiment score relies on a 0.7/0.3 text-to-emoji weighting that has not been validated against manual annotations. While alternative weighting schemes could alter individual scores, two factors mitigate this threat: we transparently report isolated VADER and emoji means throughout our results, and because automated agents rarely use emojis, their sentiment signal is functionally equivalent to the text-only baseline regardless of the ratio. Thus, our core human-versus-agent comparisons remain robust, though validating this composite scoring mechanism against a manually labeled benchmark remains an important step for future work.

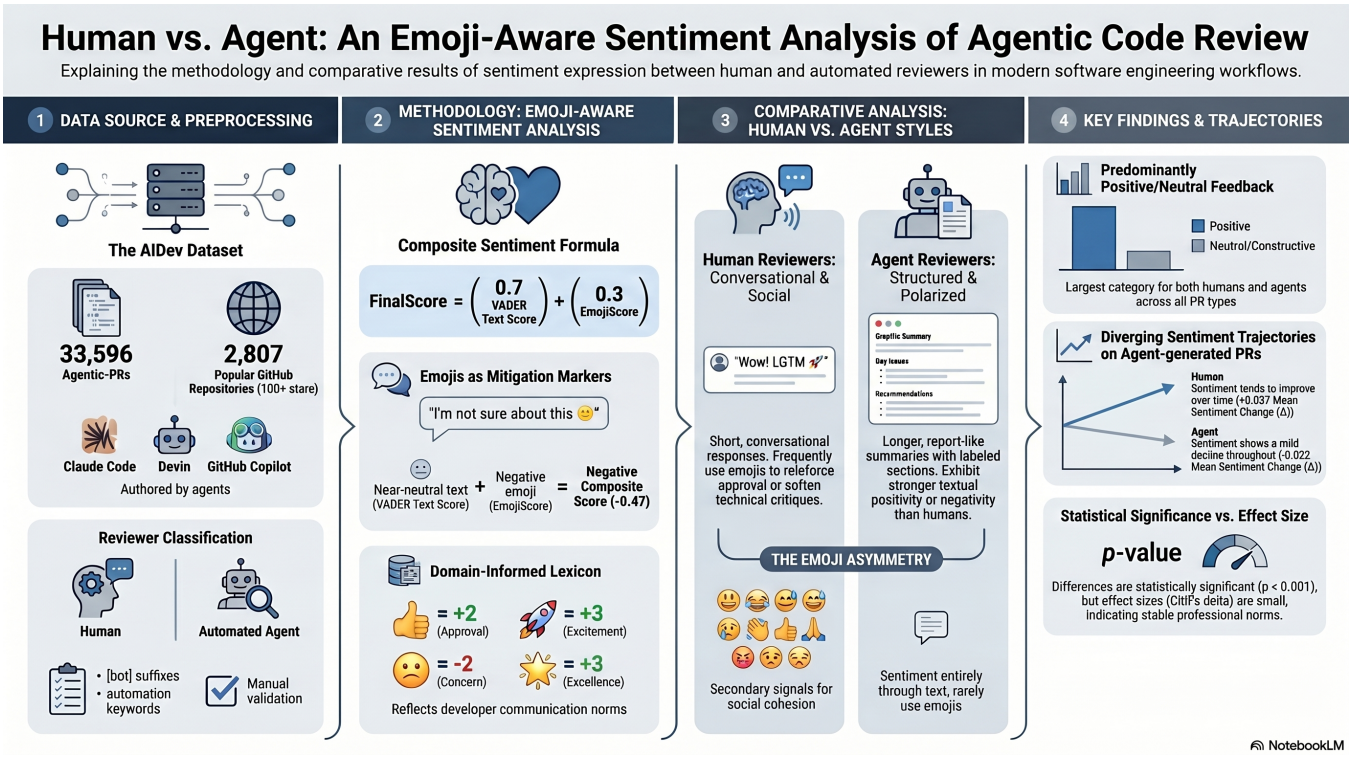

Figure 1: Summary of Methodology and Results. Image created using NotebookLM

Internal validity is limited by unmeasured confounding variables. While our design explicitly stratifies data by reviewer type and PR origin to control for primary sources of variance, we cannot establish a causal link between contribution origin and reviewer sentiment. Specifically, we did not perform a multivariate regression controlling for comment length, PR size, repository culture, agent family, or the number of review comments. While these covariates could explain part of the observed variation, their practical impact is limited: our reported effect sizes are consistently negligible-to-small ($|\delta| < 0.30$), and our overarching finding of predominantly positive or neutral sentiment holds across all experimental cohorts. Additionally, our sentiment trajectory analyses assume sequential comment ordering; if developers respond in parallel threads, computed slopes may not reflect the actual progression of the review.

External validity. We restrict our analysis to repositories with more than 100 GitHub stars (AIDev-pop), which likely represent projects with more established review practices and greater experience with automation. Results may not generalize to smaller repositories, private codebases, or projects earlier in their adoption of agent-generated contributions.

Reliability. We release the emoji lexicon, the bot classification script, and the classification rules via the replication package. VADER is a standard tool and can be applied independently. Manual validation of bot accounts introduces subjectivity, which a different team of annotators might interpret differently.

7 Limitations

While our study offers comprehensive insights into the sentiment dynamics of collaborative code review, it is subject to certain boundary conditions that limit the generalizability of our results. We highlight five primary limitations:

(1) Our dataset is restricted to GitHub repositories with over 100 stars, meaning our findings may not generalize to smaller or private commercial codebases with different review cultures.
(2) Our observational design identifies correlational sentiment patterns but precludes establishing causal links between reviewer affect and pull request origin.
(3) Our sentiment classification relies on VADER [10], which is optimized for social media and may misclassify terse or technical code review comments as neutral, likely inflating our neutral category.
(4) While our 0.7/0.3 text-to-emoji composite weighting is domain-informed and supplemented by isolated mean reporting, it has not been validated against manual human annotations.
(5) Our qualitative analysis in RQ4 serves an exploratory, illustrative purpose to provide nuance to our quantitative findings; consequently, it does not aim to establish an exhaustive taxonomy of communication styles, but rather to ground our statistical results in observable developer discourse.

8 Conclusion and Future Work

We analyzed sentiment in code review comments using the AIDev dataset and an emoji-aware approach that combines textual scores with emoji-based affective signals. Human feedback on agent-generated pull requests is predominantly positive or neutral. Emojis appear mainly as secondary signals that reinforce approval; criticism is expressed almost entirely through text. Automated reviewers exhibit stronger textual polarity in both directions but rarely use emojis, relying instead on structured, report-like comments.

Sentiment trajectories differ slightly by context: reviews of agent-generated pull requests show a mild negative slope when automated reviewers participate, whereas reviews of human-generated pull requests follow a positive trajectory regardless of reviewer type. Effect sizes are small throughout, which limits the practical significance of these differences.

Taken together, the data suggest that developers engage with agent-generated contributions in a broadly constructive way. However, the communication norms that govern these interactions differ from those observed in human-to-human review. Whether these differences reflect distrust, adaptation, or simply the functional design of automated review tools remains an open question.

As future work, interviews or surveys could triangulate these findings and clarify reviewer intent, perceived trust, and cognitive load when evaluating agent-generated code. Longitudinal studies could also examine whether sentiment patterns shift as developers accumulate experience with agent-generated contributions.

Artifact Availability

The scripts supporting the findings of this study are available at https://doi.org/10.5281/zenodo.21312971.

Use of Generative Artificial Intelligence

GenAI tools were used in this work to: (1) Gemini 3.1 Pro was used to review text and grammar and transform spreadsheets into LaTeX tables; (2) The NotebookLM tool was used to generate the image.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] Sharif Ahmed and Nasir U. Eisty. 2024. Understanding Emojis in Useful Code Review Comments. In *2024 IEEE/ACM International Workshop on Natural Language-Based Software Engineering (NLBSE)*. IEEE, New York, NY, USA, 81–84. doi:0.1145/3643787.3648035

[2] Toufique Ahmed, Amiangshu Bosu, Anindya Iqbal, and Shahram Rahimi. 2017. SentiCR: A customized sentiment analysis tool for code review interactions. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, New York, NY, USA, 106–111. doi:10.1109/ASE.2017.8115623

[3] Ikram El Asri, Noureddine Kerzazi, Gias Uddin, Foutse Khomh, and M.A. Janati Idrissi. 2019. An empirical study of sentiments in code reviews. *Information and Software Technology* 114 (2019), 37–54. doi:10.1016/j.infsof.2019.06.005

[4] Mohamed Amine Batoun, Ka Lai Yung, Yuan Tian, and Mohammed Sayagh. 2023. An Empirical Study on GitHub Pull Requests' Reactions. *ACM Trans. Softw. Eng. Methodol.* 32, 6, Article 146 (Sept. 2023), 35 pages. doi:10.1145/3597208

[5] Daniel Bee and DongGyun Han. 2025. Visualising Developer Interactions in Code Reviews. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering* (Clarion Hotel Trondheim, Trondheim, Norway) *(FSE Companion '25)*. Association for Computing Machinery, New York, NY, USA, 1070–1073. doi:10.1145/3696630.3728583

[6] Christian Bird, Denae Ford, Thomas Zimmermann, Nicole Forsgren, Eirini Kalliamvakou, Travis Lowdermilk, and Idan Gazit. 2023. Taking Flight with Copilot: Early insights and opportunities of AI-powered pair-programming tools. *Queue* 20, 6 (Jan. 2023), 35–57. doi:10.1145/3582083

[7] Daniel Coutinho, Juliana Alves Pereira, Breno Braga Neves, João Correia, Caio Barbosa, Wesley KG Assunção, Igor Steinmacher, Marco Gerosa, Augusto Baffa, and Alessandro Garcia. 2025. PRemo: A Dataset of Emotions Found on Pull Request Discussions. In *Simpósio Brasileiro de Engenharia de Software (SBES)*. SBC, SBC, Porto Alegre, RS, Brazil, 326–336. doi:10.5753/sbes.2025.9936

[8] Munmun De Choudhury and Scott Counts. 2013. Understanding affect in the workplace via social media. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work* (San Antonio, Texas, USA) *(CSCW '13)*. Association for Computing Machinery, New York, NY, USA, 303–316. doi:10.1145/2441776.2441812

[9] Emitza Guzman, David Azócar, and Yang Li. 2014. Sentiment analysis of commit comments in GitHub: an empirical study. In *Proceedings of the 11th Working Conference on Mining Software Repositories* (Hyderabad, India) *(MSR 2014)*. Association for Computing Machinery, New York, NY, USA, 352–355. doi:10.1145/2597073.2597118

[10] C. Hutto and Eric Gilbert. 2014. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media* 8, 1 (May 2014), 216–225. doi:10.1609/icwsm.v8i1.14550

[11] Hao Li, Haoxiang Zhang, and Ahmed E. Hassan. 2025. The Rise of AI Teammates in Software Engineering (SE) 3.0: How Autonomous Coding Agents Are Reshaping Software Engineering. arXiv preprint arXiv:2507.15003.

[12] Bing Liu and Lei Zhang. 2012. A Survey of Opinion Mining and Sentiment Analysis. In *Mining Text Data*, Charu C. Aggarwal and ChengXiang Zhai (Eds.). Springer US, Boston, MA, 415–463. doi:10.1007/978-1-4614-3223-4_13

[13] Xuan Lu, Yanbin Cao, Zhenpeng Chen, and Xuanzhe Liu. 2018. A First Look at Emoji Usage on GitHub: An Empirical Study. doi:10.48550/arXiv.1812.04863

[14] Arghavan Moradi Dakhel, Vahid Majdinasab, Amin Nikanjam, Foutse Khomh, Michel C. Desmarais, and Zhen Ming (Jack) Jiang. 2023. GitHub Copilot AI pair programmer: Asset or Liability? *Journal of Systems and Software* 203 (2023), 111734. doi:10.1016/j.jss.2023.111734

[15] Rajshakhar Paul, Amiangshu Bosu, and Kazi Zakia Sultana. 2019. Expressions of Sentiments during Code Reviews: Male vs. Female. In *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, New York, NY, USA, 26–37. doi:10.1109/SANER.2019.8667987

[16] Neil Perry, Megha Srivastava, Deepak Kumar, and Dan Boneh. 2023. Do Users Write More Insecure Code with AI Assistants?. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 2785–2799. doi:10.1145/3576915.3623157

[17] J. Romano, J.D. Kromrey, J. Coraggio, and J. Skowronek. 2006. Appropriate statistics for ordinal level data: Should we really be using t-test and Cohen'sd for evaluating group differences on the NSSE and other surveys?. In *annual meeting of the Florida Association of Institutional Research*. 1–3.

[18] Jaydeb Sarker, Sayma Sultana, Steven R. Wilson, and Amiangshu Bosu. 2023. ToxiSpanSE: An Explainable Toxicity Detection in Code Review Comments. In *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, New York, NY, USA, 1–12. doi:10.1109/ESEM56168.2023.10304855

[19] Rosalia Tufano, Ozren Dabić, Antonio Mastropaolo, Matteo Ciniselli, and Gabriele Bavota. 2024. Code Review Automation: Strengths and Weaknesses of the State of the Art. *IEEE Transactions on Software Engineering* 50, 2 (2024), 338–353. doi:10.1109/TSE.2023.3348172

[20] Priyan Vaithilingam, Tianyi Zhang, and Elena L. Glassman. 2022. Expectation Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 332, 7 pages. doi:10.1145/3491101.3519665

[21] Dong Wang, Tianyi Xiao, Thanh Son, et al. 2023. More than React: Investigating the Role of Emoji Reaction in GitHub Pull Requests. *Empirical Software Engineering* 28, 123 (2023), 1–25. doi:10.1007/s10664-023-10336-5

[22] J.D. Wenger, C.D. Cameron, and M. Inzlicht. 2026. People choose to receive human empathy despite rating AI empathy higher. *Communications Psychology* 4 (2026), XX–YY. doi:10.1038/s44271-025-00387-3

[23] Mairieli Wessel, Bruno Mendes de Souza, Igor Steinmacher, Igor S. Wiese, Ivanilton Polato, Ana Paula Chaves, and Marco A. Gerosa. 2018. The Power of Bots: Characterizing and Understanding Bots in OSS Projects. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 182 (Nov. 2018), 19 pages. doi:10.1145/3274451

[24] Claes Wohlin, Per Runeson, Martin Hst, Magnus C. Ohlsson, Bjrn Regnell, and Anders Wessln. 2012. *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated. doi:10.1007/978-3-662-69306-3