Judging the LLM Judges: A Human-Centric Validation of LLM-Generated Training Data for Software Retrieval

Ogtay Hasanov[1], Saad Ezzini[1,2*]
[1]Information and Computer Science Department,
King Fahd University of Petroleum and Minerals
[2]IRC for Intelligent Manufacturing and Robotics, KFUPM
Dhahran, Saudi Arabia
[*]saad.ezzini@kfupm.edu.sa

Abstract

Large Language Models (LLMs) are increasingly used to generate synthetic training data for downstream retrieval systems. While such data reduces annotation cost, low-quality generations can silently corrupt models deployed in security-sensitive settings such as software issue triage, user support, and threat-intelligence matching. We present a lightweight quality-assessment protocol for LLM-generated synthetic training data and apply it to 13,579 synthetic user reviews generated from GitHub issues across four open-source Android applications. Using a five-point alignment rubric and an independent LLM judge, we evaluate 400 stratified samples and find that 10.5% fail to meaningfully capture their source issues. Failures concentrate in developer-internal issues and sarcastic persona framings. A human-validation study with two independent annotators on 20 stratified reviews finds that the LLM judge is systematically more conservative than both humans (3-way Krippendorff's $\alpha = 0.386$; humans-only $\alpha = 0.518$), highlighting the need for hybrid human-AI verification when synthetic data is used in security-critical retrieval pipelines.

1 Introduction

LLM-generated synthetic data has become a common response to training-data scarcity. In information retrieval, recent work has shown that LLMs can generate synthetic query-document pairs that provide useful training signals for retrievers (Bonifacio et al., 2022). In software engineering, retrieval systems are used to connect user-facing reports with developer-facing artifacts, such as matching app reviews to bug reports (Häring et al., 2021). However, synthetic data can appear fluent while failing to preserve the meaning of its source artifact. When such data is used in high-stakes settings such as automated support, bug triage, or threat-intelligence matching, undetected quality degradation may propagate to silent failures in production.

We frame this as a data-integrity concern for security-sensitive retrieval pipelines, related to but distinct from adversarial poisoning: the corruption in our setting is accidental rather than malicious. Prior work shows that both training-data poisoning and retrieval-corpus poisoning can compromise downstream systems (Carlini et al., 2024; Zhong et al., 2023). In our setting, a sarcastic or generic synthetic review that fails to preserve the source issue may still teach a retriever incorrect associations.

As LLM-driven pipelines continue to scale, the need for robust evaluation frameworks has become increasingly critical. Automated metrics often fail to capture subtle semantic drifts or stylistic misalignments that can mislead downstream systems and degrade reliability. Consequently, developing human-centric validation methodologies is essential to ensure that synthetic data generation remains grounded, trustworthy, and aligned with practical use cases.

This paper makes three focused contributions. First, we propose a reproducible protocol for assessing LLM-generated training pairs using a five-point alignment rubric, stratified sampling, and an independent LLM judge. Second, we apply it to 13,579 synthetic reviews across four Android applications and score 400 stratified samples. Third, we compare LLM-based judging with human annotation by two independent raters on a 20-review subset, identify a systematic conservative bias in the LLM judge, and characterize the persona-level patterns where it diverges from human consensus.

2 Related Work

Synthetic data and software retrieval. LLM-generated synthetic data is increasingly used to construct training signals for low-label NLP and retrieval settings, but most work evaluates downstream performance rather than the fidelity of each generated instance to its source artifact (Bonifacio

Proceedings of the Second International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security (NLPAICS'2026)
11–12 June, 2026. ©2026 The Authors

et al., 2022). Work on synthetic user-generated text argues that such data should be assessed for meaning preservation, style preservation, divergence, utility, and privacy (Chim et al., 2025). In software engineering, prior work has studied matching app reviews with related bug reports (Häring et al., 2021). Our work differs by auditing whether LLM-generated reviews themselves faithfully preserve source GitHub issues before they are used as training data.

LLM judging and security risk. LLM-as-a-judge methods offer scalable evaluation for open-ended outputs (Zheng et al., 2023), and recent work studies their reliability in software engineering tasks (Wang et al., 2025). However, LLM judges can exhibit systematic biases (Wang et al., 2024); therefore, we treat LLM judging as a first-pass assessment and validate it with human annotation and agreement analysis (Cohen, 1968; Artstein and Poesio, 2008). This validation is important for security-sensitive retrieval pipelines, where corrupted training data or retrieval corpora can silently distort system behavior (Carlini et al., 2024; Zhong et al., 2023).

3 Quality Assessment Protocol

Our protocol is designed for synthetic training pairs where a generated text instance should preserve the meaning of a source artifact.

3.1 Alignment Rubric

For each synthetic instance, a judge rates alignment with the source artifact on a 1–5 scale:

- **5**: Accurately captures the specific issue and key technical details.
- **4**: Captures the core problem with minor vagueness.
- **3**: Relates to the general topic but lacks specificity.
- **2**: Has only a vague connection to the issue.
- **1**: Has no meaningful connection.

The rubric operationalizes source-review fidelity for synthetic user-generated text. It is inspired by prior work on synthetic text evaluation and software artifact matching (Chim et al., 2025; Häring et al., 2021), but is tailored to judging whether a generated user review preserves the meaning of its source GitHub issue. Because the scale is ordinal, we use weighted agreement measures when comparing raters (Cohen, 1968; Artstein and Poesio, 2008).

3.2 Stratified Sampling and Judge Model

We sample across application and persona. Application-level stratification surfaces domain-specific failures, while persona-level stratification surfaces generation-style failures. Within each stratum, uniform random sampling is used.

To avoid judge-generator contamination, we use GPT-4o-mini as the judge for data generated by Mistral. Prior work suggests that LLM judges can align with human evaluators in some open-ended and software engineering settings (Zheng et al., 2023; Wang et al., 2025), but also that they can exhibit systematic biases (Wang et al., 2024). We therefore treat the judge as a scalable first-pass assessor rather than ground truth.

4 Case Study: Software Issue Reviews

4.1 Setup

We apply the protocol to synthetic user reviews generated for a software issue retrieval system. Mistral was prompted to assume one of ten user personas and produce a colloquial review describing a real GitHub issue from four open-source Android applications: Brave Browser, Signal, AnkiDroid, and K-9 Mail. In total, 13,579 synthetic reviews were generated. We sampled 100 reviews per application (400 total), preserving uniform persona coverage, and rated each review using GPT-4o-mini following the rubric in Section 3.1.

4.2 Results

Across the 400 sampled reviews, 267 (66.8%) were correct (scores 4–5), 91 (22.8%) were partially correct (score 3), and 42 (10.5%) were weak (scores 1–2), with a mean score of 3.56/5.0. Thus, roughly one in ten synthetic reviews failed to meaningfully capture its source issue.

Table 1 shows that Signal produced the highest-quality synthetic reviews, while AnkiDroid produced the most weak reviews. Manual inspection shows that many weak AnkiDroid cases correspond to developer-internal issues such as refactoring, continuous integration, or build-system updates. Such issues have no obvious user-visible symptom, forcing the generator to fabricate plausible complaints.

App	Mean	Correct	Partial	Weak
Signal	3.74	80	14	6
K-9 Mail	3.59	66	27	7
Brave	3.46	59	28	13
AnkiDroid	3.46	62	22	16

Table 1: Quality by application (n=100 per app). Correct=4–5, Partial=3, Weak=1–2.

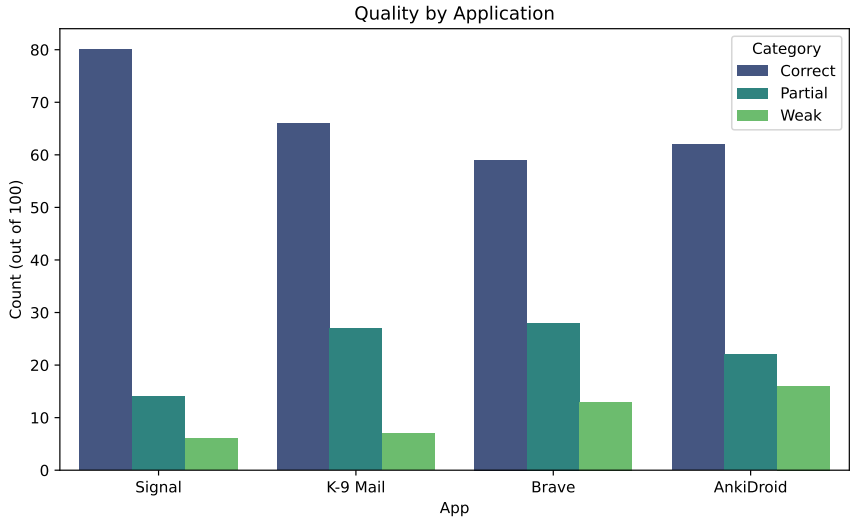

Figure 1: Visualizing the quality distribution of synthetic reviews by application, based on Table 1.

Scores also vary across personas: *detailed_complaint* (3.93), *calm_reporter* (3.80), *technical_user* (3.79), *comparison_user* (3.74), *broken_english* (3.64), *desperate_user* (3.62), *one_liner* (3.41), *confused_user* (3.40), *angry_short* (3.21), and *sarcastic* (3.10). Sarcastic and angry-short personas were weakest because they often substituted attitude for source-specific detail.

5 Human Validation

To validate the LLM judge, we sampled 20 reviews (five per application) and had two independent human raters score them using the same 1–5 rubric: the first rater is one of the authors, and the second is a graduate student who annotated the subset blind to both the LLM scores and the first rater's scores.

Table 2 reports pairwise agreement. Across all three raters, Fleiss' $\kappa = 0.247$ (nominal) and Krippendorff's $\alpha = 0.386$ (ordinal, 95% bootstrap CI: [0.03, 0.67]). When restricted to the two human raters alone, ordinal α rises to 0.518, meaning the LLM judge — not either human — is the outlier in the three-way comparison.

The mean scores reveal a systematic directional bias: H1 = 4.25, H2 = 3.85, LLM = 3.05. Both humans score these reviews higher than the LLM, with mean human–LLM gap of $+1.20$ (H1) and $+0.80$ (H2). Among the 10 items where the two humans scored identically, the LLM scored *lower* in 5 cases, equal in 5, and never higher. Excluding ties, this yields five lower-than-human cases and zero higher-than-human cases; under a null hypothesis that the judge is equally likely to over-score or under-score relative to human consensus, a one-sided sign test gives $p = 0.031$. This suggests that the disagreement is directional rather than purely noisy.

Manual inspection of the largest divergences reveals two patterns. First, the LLM penalizes missing technical specifics (e.g., app version, device model) where human raters award credit for correct topic alignment. Second, sarcastic reviews that preserve issue-relevant content receive substantially higher human than LLM scores: a Signal review about a duplicate-backup bug delivered in sarcastic tone, for instance, is scored 4 and 5 by the two humans but 1 by the LLM, despite preserving the core technical content. The persona-level pattern in the validation subset replicates the case study: mean human–LLM gap is largest for *sarcastic* ($+1.67$), *confused_user* ($+1.50$), and *angry_short* ($+1.17$)

Pair	κ_{lin}	95% CI	ρ	Exact	≤ 1
H1–H2	0.442	[0.10, 0.71]	0.560**	50%	85%
H1–LLM	0.286	[0.06, 0.53]	0.500*	35%	70%
H2–LLM	0.477	[0.20, 0.73]	0.574**	50%	85%

Table 2: Pairwise inter-rater agreement on 20 stratified reviews. κ_{lin}: linear-weighted Cohen's κ; 95% CIs from 10,000 bootstrap resamples. ρ: Spearman rank correlation. $^*p < 0.05$, $^{**}p < 0.01$.

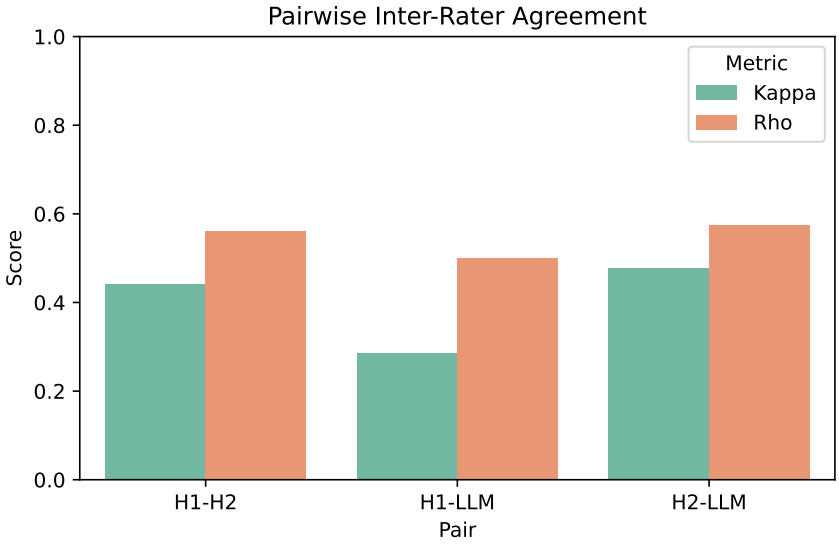

Figure 2: Visualization of pairwise inter-rater agreement metrics from Table 2.

— the same persona families flagged as weakest in Section 4.2. These results indicate that LLM-only judging systematically underestimates synthetic-data quality on this rubric, particularly for stylistically marked personas, motivating hybrid protocols where topic-level fidelity is the criterion of interest.

6 Discussion and Conclusion

Our case study shows that LLM-generated synthetic training data has non-trivial quality degradation: 10.5% of generated instances fail to meaningfully capture their source artifacts, with failures concentrated in predictable categories such as developer-internal source material and sarcastic persona framings. For security-critical retrieval systems, where an incorrectly trained retriever can misroute user queries to wrong bug reports or threat-intelligence entries, this level of silent data-quality degradation can be consequential.

The human-validation results further indicate that automated quality assessment alone is insufficient: the LLM judge is systematically more conservative than both human raters and never overscores on items where the humans agreed. Therefore, the absolute 10.5% weak rate should be interpreted cautiously, since judge choice may affect the estimated prevalence of weak samples, especially for stylistically marked personas. LLM judges remain useful for scalable screening, but their scoring underestimates topic-level fidelity in systematic, persona-correlated ways. We therefore recommend hybrid protocols that combine LLM-scale scoring with targeted human verification of edge cases, especially low-specificity outputs, sarcastic framings, and source issues with no clear user-visible symptom.

Limitations. This work studies one domain, one generator model, and one LLM judge. The 20-review validation subset yields wide bootstrap CIs on agreement statistics: while the *direction* of the LLM judge's bias is consistent across both human raters and matches the persona-level pattern observed in the 400-sample case study, the *magnitude* of inter-rater agreement is uncertain at this sample size. Future work should evaluate additional generators, judges, domains, and larger human-annotated samples.

Ethics Statement. This work uses publicly available GitHub issue data and synthetic reviews gen-

erated from that data. No personal user data is involved. Human annotators participated voluntarily and were informed of the study purpose. We will release the protocol, sampling code, and annotation data upon publication to support reproducibility.

Acknowledgments

We would like to thank our colleagues and reviewers for their valuable feedback and support in improving this work. We also gratefully acknowledge the open-source community for providing the resources and tools that made this research possible.

References

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. InPars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392.

Nicholas Carlini, Matthew Jagielski, Christopher A. Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. 2024. Poisoning web-scale training datasets is practical. In *Proceedings of the 2024 IEEE Symposium on Security and Privacy*, pages 407–425.

Jenny Chim, Julia Ive, and Maria Liakata. 2025. Evaluating synthetic data generation from user generated text. *Computational Linguistics*, 51(1):191–233.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220.

Marlo Häring, Christoph Stanik, and Walid Maalej. 2021. Automatically matching bug reports with related app reviews. In *Proceedings of the 43rd International Conference on Software Engineering*, pages 970–981.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 9440–9450.

Ruiqi Wang, Jiyu Guo, Cuiyun Gao, Guodong Fan, Chun Yong Chong, and Xin Xia. 2025. Can LLMs replace human evaluators? an empirical study of LLM-as-a-judge in software engineering. *Proceedings of the ACM on Software Engineering*, 2(ISSTA).

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36.

Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. 2023. Poisoning retrieval corpora by injecting adversarial passages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13764–13775.