Can LLMs Replace Manual Annotation of Software Engineering Artifacts?

Tongwei Zhang
t98zhang@uwaterloo.ca
University of Waterloo
Waterloo, Canada

ACM Reference Format:
Tongwei Zhang. 2025. Can LLMs Replace Manual Annotation of Software Engineering Artifacts?. In *Proceedings of Please replace with your conference (Conference acronym 'XX)*. ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 What is the Problem Being Solved

Human-subject evaluations are important in software engineering research, for example to judge the quality of code summaries, detect bugs, or assess static analysis warnings. But they are slow and expensive, especially when using professional developers at market rates. Using students is cheaper but may not generalize well. Many tasks require multiple ratings per artifact for reliability, which increases cost further. With large language models (LLMs) now showing strong performance on SE tasks, the question is when and how they can safely replace some human annotation effort without harming reliability.

2 What is the New Idea

The paper presents the first systematic study of LLMs as substitutes for human annotators in SE. The authors apply six state-of-the-art LLMs to ten annotation tasks from five datasets, covering code summarization, name–value consistency, semantic similarity, causality detection, and static analysis warnings. They compare human–human, human–model, and model–model inter-rater agreement.

They find that model–model agreement correlates strongly with human–model agreement, suggesting it can be used as a cheap predictor of whether a task is suitable for LLM substitution. They also use an LLM's output probability as a confidence score to identify specific samples where the model is likely to match human judgment. This enables partial replacement of human ratings while keeping statistical reliability.

In many tasks, replacing one human rater with an LLM for 50–100% of samples preserves agreement levels, leading to potential savings of up to 33% of total annotation effort. Based on these findings, they propose a decision workflow: (1) query multiple strong LLMs with few-shot prompts for all samples; (2) if model–model agreement exceeds 0.5, replace one human rating per sample with LLM output; (3) if lower, replace only high-confidence samples.

3 Positive Points

Clear, evidence-based methodology. The study spans 10 tasks from 5 datasets, with careful comparison of human–human, human–model, and model–model agreement, making the conclusions more credible.
Actionable decision framework. The two-step workflow using model–model agreement and model confidence is practical and easy to apply in real research settings.
Quantified effort savings. The paper estimates concrete savings (up to 33%) while maintaining reliability, which is directly useful for planning SE studies.

4 Negative Points

Limited task diversity. All tasks are discrete-label annotation problems; results may not generalize to open-ended or exploratory annotation tasks.
Training data leakage not fully addressed. Some datasets may be in LLM training corpora, but the mitigation discussion is brief.
No developer-centric validation. The study measures agreement but not whether LLM-assisted annotations improve downstream developer decisions.

5 Future Work

Developer-impact study. Integrate LLM-assisted annotations into real workflows (e.g., code review, bug triage) and measure productivity, accuracy, and satisfaction.
Adaptive human–LLM collaboration. Build a live annotation platform that routes tasks to humans or LLMs based on real-time model–model agreement and confidence.

6 Rating

4 out of 5. A strong, well-validated framework for reducing annotation costs in SE research, though currently limited to certain task types.

7 Discussion Points

Reliability vs. utility. Does matching human agreement ensure the annotations are actually useful for improving tools or processes?

- *Measuring reliability.* We discussed how to actually measure reliability in a fair way. The paper uses several agreement metrics, but reliability itself is fuzzy because even human–human agreement is not perfect and can be biased. One idea from class is to add a structured human feedback loop

Unpublished working draft. Not for distribution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference acronym 'XX, Your City, Country
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2025/06
https://doi.org/XXXXXXX.XXXXXXX

as a complementary check, so that we do not rely only on agreement numbers.

- *Systematic errors.* If both humans and LLMs are wrong in the same direction, they may agree but still be incorrect. So agreement alone cannot guarantee correctness. This is a real risk when tasks are ambiguous or the dataset has hidden bias.
- *Reliability before utility.* Some classmates felt it is still too early to push on "utility" in practice. Right now the indicators for usefulness are not very explicit or robust, so we should secure reliability first before claiming strong utility in real SE workflows.
- *Task-dependent usefulness.* The usefulness of LLMs probably depends on the specific annotation task. A single universal measure of "utility" looks unrealistic. Designing fair and general evaluation is costly and also technically heavy, which reduces the simplicity story.
- *Cost and implementation difficulty.* Large-scale deployment needs multiple strong models, careful prompts, and infrastructure. This is expensive and hard to maintain. In practice, these costs can offset part of the savings from reducing human annotation effort.
- *Broader vision vs. immediate practice.* The professor reminded us that many papers, including this one, are still more like a vision for how LLMs could support SE in the near future. The path from this vision to day-to-day deployment is not short.

Ethics and bias. Could replacing humans with LLMs amplify biases in SE datasets, and how can this be detected and mitigated?

- *Bias and fairness.* Both humans and LLMs carry bias. Since LLMs learn from large human-generated data, they may repeat and even amplify unfair patterns. This includes discrimination risk if we are not careful with datasets and prompts.
- *Compliance and accountability.* We discussed possible legal and regulatory pressure in the future. Researchers may need to show that training data and annotation pipelines are sufficiently unbiased, which is not trivial to prove.
- *Vulnerability to misuse.* There is a risk of adversarial attacks or malicious use. If biased or manipulated annotations flow into SE tools, the downstream impact can be negative and persistent.
- *Lessons from other domains.* A classmate compared with high-stakes areas like medicine or drug development, where ethical and legal consequences of bias are much stronger. This suggests our community should prepare guardrails early, even if SE is not always high-stakes.

8 Other Comments

The work is timely and could influence how SE studies are run, especially for large-scale annotation tasks. In one of my recent project, I can leverage this one as a reference to prove the reliability of my LLM-annotation strategy.

Unpublished working draft. Not for distribution.