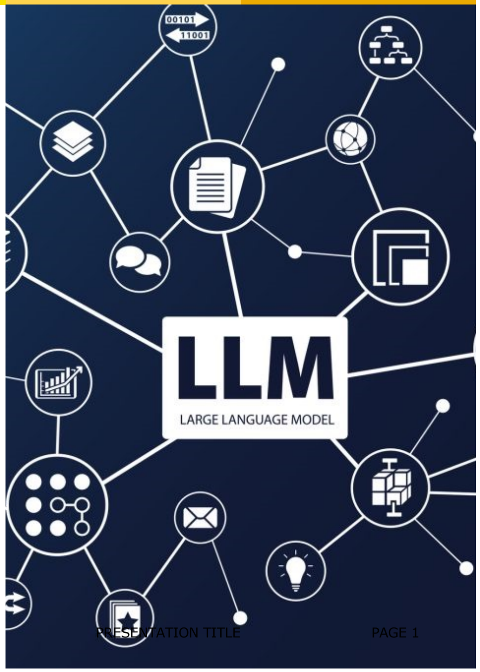
CAN LLMS REPLACE MANUAL ANNOTATION OF SOFTWARE ENGINEERING ARTIFACTS?

Presented by Tongwei Zhang

UNIVERSITY OF WATERLOO



Background of the AI4SE generation (from textbook)

- **Naturalness of software**
  - Software exhibits simplicity and predictability akin to natural language.
  - This opens the door to **statistical/NLP methods** for SE tooling and practices.
- **Software as human-produced text**
  - Shares many statistical properties with natural language.
  - Enables numerous **assistive SE applications** (e.g., analysis, automation).
- **Additional context**
  - **NLP in SE** and **software repository mining** connect theory → practical tools that help engineers.
  - **LLMs now participate** in solving these problems.

- **However → Core question:** *How much effort should LLMs contribute to SE evaluations?*

Problem to be solved by this paper

- Human-subject evaluations are expensive & slow
  - Needed for: code summarization, bug detection, static analysis usefulness, etc.
  - Recruiting professional developers: costly (e.g., ~$60/hour) and time-intensive.
  - Using students risks poor generalizability.
  - Multiple ratings per artifact required for reliability → costs multiply.

- Need scalable, reliable alternatives
  - LLMs show strong SE task performance.
  - Challenge: When and how can LLMs safely substitute humans?

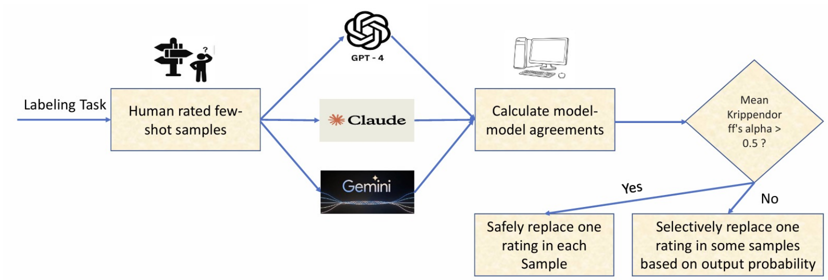
New Ideas (Empirical)

- First systematic study of LLMs as human annotation substitutes in SE
  - 6 state-of-the-art LLMs × 10 tasks × 5 datasets
  - Tasks: code summarization, name–value consistency, semantic similarity, causality detection, static-analysis warnings
  - Compare H2H, H2M, M2M inter-rater agreement
- Model–model agreement (M2M) as predictor
  - Strong correlation between M2M and H2M
  - Use M2M (cheap to compute) to decide task suitability for LLM substitution.

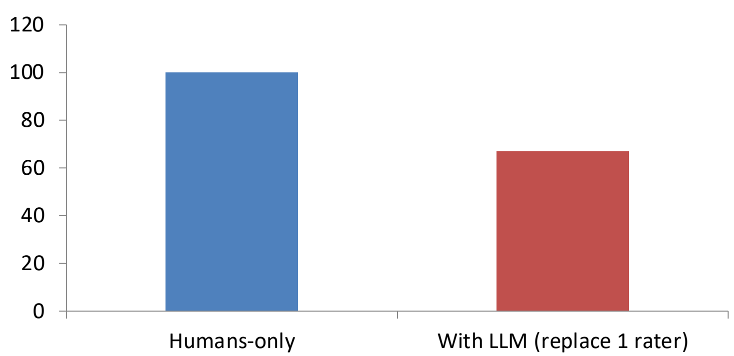
New Ideas (Methodological)

- Model confidence for sample-level selection: use model probability to pick safe samples
- Efort-saving strategy: replace one human for 50–100% (up to ~33% savings)
- Proposed decision workflow: Step1 M2M; Step2 >0.5 replace; Step3 else high-confidence only

VISUALIZE DECISION FLOW



Effort Savings (Illustrative)



Illustrative: replacing one of three raters ≈ 33% less human effort; agreement preserved where policy applies.

Positives

- Clear, evidence-based methodology
  - 10 tasks, 5 datasets; H2H/H2M/M2M comparisons → credibility across contexts.
- Actionable decision framework
  - Policy: M2M for suitability, confidence for sample selection.
- Quantified effort savings
  - Concrete potential savings (up to 33%) while preserving reliability.

Negatives

- Task diversity limited
  - All tasks are discrete-label annotations; generalization to open-ended/qualitative tasks unclear.
- Training data leakage not fully addressed
  - Public datasets may be in pretraining; mitigation discussion is brief.
- No developer-centric validation
  - Agreement measured, but not whether LLM-assisted annotations improve downstream developer decisions.

Future Work

- Developer-impact study
  - Integrate LLM-assisted annotations into real workflows (code review, bug triage); measure productivity, accuracy, satisfaction.
- Adaptive human–LLM collaboration system
  - Live platform computing M2M + confidence in real time; route tasks dynamically; test scalability in production.

Rating

- **4 (Very Strong Contribution)**
- A well-validated, actionable framework for reducing annotation costs in SE research, with scope currently limited to certain task types.

Discussion Points

- Reliability vs. Utility:
  - If LLMs match human agreement, do annotations actually improve developer tools/processes? How to measure usefulness beyond agreement?
- Ethics & Bias:
  - If LLMs inherit biased data, could substitution amplify biases in SE datasets? How to detect & mitigate?

UNIVERSITY OF WATERLOO

YOU+WATERLOO

Our greatest impact happens together.