Mittuniversitetet
MID SWEDEN UNIVERSITY

Exploring the Use of Large Language Models for Data Extraction for Systematic Reviews in Software Engineering

Muhammad Laiq

muhammad.laiq@miun.se

IARIA

Muhammad Laiq

Muhammad Laiq is a **Postdoctoral** Researcher in Software Engineering at Mid Sweden University. He recently (Feb 2025) completed his PhD in Software Engineering from **Blekinge Institute of Technology**. His dissertation investigates and evaluates the practical application of data-driven approaches (including traditional machine learning and large language models) in industrial settings to assist practitioners in defect management.

His broader research interests include empirical software engineering, software quality, and AI-driven software engineering.

Introduction

- Systematic reviews are vital for evidence-based decision-making in software engineering.
- Data extraction is a key but labor-intensive step for systematic reviews.
- Large language models (LLMs), e.g., GPT-based models, can help reduce manual effort.
- **Gap:** There is limited research on the use of LLMs for systematic reviews in software engineering.

Research objective

- Evaluate GPT-4o's performance in data extraction for a systematic mapping study (SMS).
- Compare the performance of GPT-4o with manually extracted data.

Replicated study

- SMS focused on issue report classification, i.e., classify reports as bugs or non-bugs.
- 46 primary studies included in replication.

Research method (1/2) – Prompt and data extraction template

Question	**Prompt description:** "You have been provided 46 papers on issue report classification in software engineering. Your task is to extract data from the provided papers using the following data extraction template." **"Data extraction template"** **"Item ID. Description"**
RQ1	1. Proposed automatic techniques for classification, e.g., Logistic regression and RoBERTa.
RQ2	2. Used features, e.g., title, description, body, and priority of an issue report.
RQ3	3. Used pre-processing techniques (or a tokenizer) for feature extraction from textual features, e.g., Word2vec, TF-IDF, and BERT-based tokenizer.
RQ4	4. Study context, i.e., data from Open-Source (OSS) or Closed-Source (CSS) that was used in the study.
RQ5	5. Does the study involve practitioners for feedback? Yes or No.

Research method (2/2) – Assessment criteria

Score	Assessment criteria
1	If all identified items by GPT are correct.
0.75	If more than half of the correct items have been identified.
0.5	If half of the correct items have been identified.
0.25	If less than half of the correct items have been identified.
0	If none of the identified items are correct.
	Score for RQ = Sum of score for all studies / Maximum score (46)

Results

Question	Score of GPT-4o
RQ1	75% (34.25/46)
RQ2	77% (35.5/46)
RQ3	64% (29.25/46)
RQ4	98% (45/46)
RQ5	84% (38/46)
	Overall score = 79% ((34.25/46) + (35.5/46) + (29.25/46) + (45/46) + (38/46)) / 5

Conclusion

- Our evaluation revealed that GPT-4o achieves an average accuracy of approximately 79%.
- Although these results indicate that the entire process cannot be fully automated, GPT-4o can be a supportive tool in a semi-automated workflow.
- Therefore, we recommend using LLMs, such as GPT-4o, for an initial phase of automated extraction, followed by human validation and refinement.

Future work

- Future work will focus on exploring several other LLMs (e.g., the models from Gemini, Llama, and DeepSeek), including their evaluation in other areas of SE, e.g., effort estimation, code quality, and defect prediction.

Thank you!