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

Muhammad Laiq
Department of Communication, Quality Management and Information Systems
Mid Sweden University
Campus Östersund, Sweden
e-mail: muhammad.laiq@miun.se

***Abstract*—To support evidence-based decision-making, software engineering employs systematic reviews to collect and consolidate relevant literature on a specific research topic. However, conducting systematic reviews is a labor-intensive and time-consuming task. Recent advancements in Large Language Models (LLMs), such as Generative Pre-trained Transformer (GPT) models, offer opportunities to streamline and reduce the manual effort required, particularly in data extraction for Systematic Mapping Studies (SMS). This study evaluates the performance of GPT-4o in extracting data from 46 primary studies of an SMS by comparing the results of automated extraction with the data extracted manually. 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.**

Keywords-LLMs; Data extraction; Systematic mapping study; literature review; Systematic reviews.

I. Introduction

In Software Engineering (SE), systematic reviews, including systematic literature reviews, systematic mapping studies [1], and tertiary studies [2], are commonly used to aggregate evidence on a particular topic. A number of systematic reviews have been performed in almost all areas of SE, e.g., [3]–[6]. Conducting these reviews requires significant effort as they follow a rigorous process that includes several steps, including identifying the need for a review, defining a search strategy, defining selection criteria, selecting relevant studies, and extracting required data. Among these steps, data extraction is an important and effort-intensive step, and has been done manually so far. In addition, this step has received the least attention regarding automated support for conducting systematic reviews [7][8].

Recent advances in Large Language Models (LLMs) have led to increasing attention to automating the data extraction process for systematic reviews [9]–[12]. However, there is still a lack of such attempts in SE. In their recent work, Felizardo et al. [7] were the first to evaluate an LLM for data extraction for a systematic mapping study in the SE area. Their results indicate that LLM-based tools could be a promising solution to assist with data extraction in conducting systematic reviews. However, they stressed the need for further research (evidence) in the SE domain before this technology can be adopted. Building on their work, we contribute in this direction by evaluating an LLM for data extraction for a systematic mapping study [13].

In this study, we evaluate the performance of GPT-4o in extracting data from 46 primary studies for a systematic mapping study [13]. We compared the results of the data extracted automatically using GPT-4o with those obtained through manual extraction. Our evaluation shows that GPT-4o achieves an average accuracy of approximately 79%.

Our overarching goal is to evaluate the ability of LLMs in the data extraction step to conduct systematic reviews in SE. This paper outlines the current status of our ongoing efforts to achieve this goal. Future work will explore several other LLMs (such as models from Gemini, Llama, and DeepSeek) and their evaluation in other areas of SE, including effort estimation, code quality, and defect prediction.

The remainder of the paper is organized as follows. Section II provides background information about the task studied and the replicated mapping study. Section III presents the research method of our study. Section IV presents the results of our study. Section V discusses the study findings, describes related work on the topic, and discusses potential validity threats to the study. Finally, Section VI concludes this study with future work.

II. Background: Task and replicated SMS

In this section, we provide background information about the task studied (that is, conducting a systematic mapping study in software engineering) and the replicated mapping study.

A. Task: Conducting a systematic mapping study

Systematic Mapping Studies (SMS), also known as scoping studies, aim to provide a comprehensive overview of a specific research area by categorizing and quantifying existing literature [14]. SMS focuses on structuring a field by identifying what has been studied, the methodologies used, and where the results have been published. These studies help identify research trends, gaps, and opportunities for future research.

Conducting an SMS is a rigorous and resource-intensive process that involves several essential steps. These steps include identifying the need for the study, designing a search strategy, defining inclusion and exclusion criteria, selecting relevant primary studies, and extracting and analyzing data from the chosen studies. Among these steps, data extraction is particularly laborious and has traditionally been performed

manually. Despite its importance, it remains the least supported step regarding automation tools. With the advent of LLMs, there is a growing interest in leveraging these technologies to assist with data extraction in SMSs. In particular, Felizardo et al. [7] were the first to explore using an LLM for data extraction in the SE context. Their findings indicated that LLMs show promising results, but highlighted the need for further evidence before adopting them in SE research workflows. This study builds on their work by evaluating the use of GPT-4o for automated data extraction for an SMS in SE.

B. Replicated SMS

To evaluate the effectiveness of GPT-4o for data extraction, we replicate the data extraction step for a manually conducted systematic mapping study on issue report classification [13]. In SE, the goal of issue report classification is to support effective defect management by categorizing reported issues early on into [13][15][16]: (a) Bugs: Issues that require code changes or fixes, and (b) Non-bugs: Including feature requests, questions, and documentation issues. This classification helps practitioners prioritize and assign resources more efficiently during software maintenance and evolution. In this replication, we use GPT-4o to extract data from 46 primary studies included in the original SMS on issue report classification [13]. We aim to evaluate the accuracy of GPT-4o in automating the data extraction process for an SMS.

III. Methodology

As a first step towards achieving our overarching goal of evaluating the ability of LLMs to extract data for systematic reviews in SE, we conducted an initial proof-of-concept study. For this assessment, we chose a systematic mapping study focused on classifying software issue reports [13]. We extracted data related to the five research questions listed in Table I for the selected study.

In this proof-of-concept study, we selected GPT-4o as an LLM to evaluate its data extraction performance for the chosen systematic mapping study in SE. We will compare the performance of GPT-4o with data manually extracted by human researchers.

Table I shows the template with the instructions we used to extract data using GPT-4o. At first, we provide GPT-4o with the set of all the papers as PDFs. PDFs for 46 primary studies were provided in batches. Then, GPT-4o was prompted to extract the data from these PDFs using the template.

In Table II, we describe our assessment criteria for evaluating the responses of GPT-4o. We evaluated the responses of GPT-4o against the manually extracted data items as follows. The responses of GPT-4o are compared with the ground truth by the authors. A score of 1 (the maximum) is awarded if all the items identified by GPT-4o are correct. We assign a score of 0.75 if more than half of the correct items are identified, 0.5 if exactly half are identified, and 0.25 if less than half are identified. If none of the identified items are correct, the score will be 0. To calculate the final score for each research question, we use the following formula: **Sum of scores for all studies / maximum score (46)**.

TABLE I. PROMPTS 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.

TABLE II. ASSESSMENT CRITERIA APPLIED TO EXTRACTED DATA FOR EACH RQ USING GPT-4O

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)

IV. Results

In this section, we present the evaluation results, that is, the performance of GPT-4o for the data extraction for the systematic mapping study on software issue report classification [13]. Table III presents the results of GPT-4o. The model achieved an overall accuracy of 79% in extracting all data items (RQ). The performance of GPT-4o varied across different questions, with the highest accuracy, 98%, recorded for RQ4, indicating a near-perfect extraction performance for that particular question. For RQ5 and RQ2, GPT-4o also demonstrated promising results, achieving scores of 84% and 77%, respectively. In contrast, RQ1 received a score of 75%. The lowest performance was observed for RQ3, which scored 64%, suggesting that this question posed more challenges for GPT-4o. Overall, these results indicate that GPT-4o can support data extraction for most aspects of the mapping study, although there is some variability between questions.

V. Discussion and validity threats

In this section, we discuss the findings of the study, describe related work on the topic, and discuss potential validity threats to the study.

TABLE III. GPT-4O PERFORMANCE FOR DATA EXTRACTION FOR THE SYSTEMATIC MAPPING STUDY FOR EACH RQ

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

A. Discussion

Recent advances in LLMs have led to increasing attention to automating the data extraction process for systematic reviews [9]–[12]. However, there is still a lack of such attempts in SE. Felizardo et al. [7] reported that their work is the first attempt in the context of SE. They reported that their results indicate that LMM-based tools can be a promising solution to assist in data extraction for systematic reviews in SE. However, they emphasized that more research is needed in the context of SE. Building on their work, we contribute in this direction by evaluating GPT-4o for data extraction for a systematic mapping study on software issue report classification [13].

Our findings indicate that GPT-4o can achieve an average accuracy of 79%. This suggests that GPT-4o can assist in data extraction for systematic mapping studies in SE. However, with these results, researchers should consider using a hybrid (semi-automated) approach that combines automated extraction with manual verification to ensure the reliability of systematic reviews. For example, a semi-automated workflow could begin using an LLM, such as GPT-4, to extract data from a small sample of primary studies. The initial outputs would then be manually validated against the ground truth to assess the model's performance and identify common errors. Based on this review, researchers can refine prompts and add more contextual information to better align with the structure and semantics of the target data. This iterative feedback loop will help tailor the LLM's behavior to the specific context of an SMS, ultimately improving the extraction quality before scaling to the full dataset.

B. Validity threats

Our results are based on a single systematic mapping study that included only 46 primary studies. Furthermore, we have evaluated only a single LLM, i.e., GPT-4o. These are the limitations of our study. Additional studies evaluating more models in various areas of SE are needed for more generalizable findings.

In this study, we also relied on human-extracted data as a benchmark for our analysis, which could introduce potential human error and bias, which could affect the accuracy of our findings. However, we consider this threat minimal since two researchers were involved in the data extraction process for the selected mapping study [13]. Additionally, we acknowledge that the responses generated by GPT may vary due to its stochastic nature, which may have influenced our results.

VI. Conclusion and Future Work

In this study, we evaluated the performance of GPT-4o in data extraction for a systematic mapping study in SE. We compared its performance against data extracted manually from 46 primary studies. Our results indicate that GPT-4o can achieve an average accuracy of approximately 79%, demonstrating its potential as a valuable tool in the data extraction process. Although it cannot completely replace the manual approach, incorporating GPT-4o into a semi-automated workflow can significantly improve efficiency. We recommend starting with LLMs for automatic data extraction, followed by human review and refinement of the results. This combination will likely reduce effort while ensuring that the extracted data remains accurate and reliable.

Our overarching goal is to evaluate the ability of LLMs to assist in the data extraction step for conducting systematic reviews in SE. This paper presents the current status of our ongoing attempt towards achieving this goal. 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.

In this work, we manually validated the responses generated by the LLM against the ground truth data. Although this approach provides a qualitative understanding of the model's performance, it is inherently subjective and labor-intensive. As part of future work, we plan to adopt more automated and objective evaluation methods. In particular, we aim to incorporate automated metrics, such as n-gram or LLM itself as a judge, to compare LLM-extracted data with manually curated ground truth objectively. This will enable a more rigorous and reproducible assessment of the LLM's performance and facilitate comparison across studies.

In addition, future work will also systematically explore the impact of different prompt engineering techniques on the accuracy and reliability of LLMs for data extraction for systematic reviews.

References

[1] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-based software engineering and systematic reviews*. CRC press, 2015.

[2] B. Kitchenham *et al.*, "Systematic literature reviews in software engineering–a tertiary study", *Information and software technology*, vol. 52, no. 8, pp. 792–805, 2010.

[3] R. Hoda, N. Salleh, J. Grundy, and H. M. Tee, "Systematic literature reviews in agile software development: A tertiary study", *Information and Software Technology*, vol. 85, pp. 60–70, 2017.

[4] S. Zein, N. Salleh, and J. Grundy, "Systematic reviews in mobile app software engineering: A tertiary study", *Information and Software Technology*, vol. 164, p. 107 323, 2023.

[5] D. Budgen and P. Brereton, "Evolution of secondary studies in software engineering", *Information and Software Technology*, vol. 145, p. 106 840, 2022.

[6] M. Laiq, N. b. Ali, J. Börstler, and E. Engström, "Software analytics for software engineering: A tertiary review", *arXiv preprint arXiv:2410.05796*, 2024.

[7] K. R. Felizardo *et al.*, "Data extraction for systematic mapping study using a large language model-a proof-of-concept study in software engineering", in *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2024, pp. 407–413.

[8] K. R. Felizardo and J. C. Carver, "Automating systematic literature review", *Contemporary empirical methods in software engineering*, pp. 327–355, 2020.

[9] Z. Sun *et al.*, "How good are large language models for automated data extraction from randomized trials?", *medRxiv*, pp. 2024–02, 2024.

[10] G. Gartlehner *et al.*, "Data extraction for evidence synthesis using a large language model: A proof-of-concept study", *Research synthesis methods*, vol. 15, no. 4, pp. 576–589, 2024.

[11] M. P. Polak and D. Morgan, "Extracting accurate materials data from research papers with conversational language models and prompt engineering", *Nature Communications*, vol. 15, no. 1, p. 1569, 2024.

[12] S. A. Mahuli, A. Rai, A. V. Mahuli, and A. Kumar, "Application chatgpt in conducting systematic reviews and meta-analyses", *Br Dent J*, vol. 235, no. 2, pp. 90–92, 2023.

[13] M. Laiq and F. Dobslaw, "Automatic techniques for issue report classification: A systematic mapping study", *arXiv preprint arXiv:2505.01469*, 2025.

[14] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering", in *12th international conference on evaluation and assessment in software engineering (EASE)*, BCS Learning & Development, 2008.

[15] M. Laiq, N. bin Ali, J. Börstler, and E. Engström, "A comparative analysis of ml techniques for bug report classification", *Journal of Systems and Software*, p. 112 457, 2025.

[16] G. Antoniol, K. Ayari, M. Di Penta, F. Khomh, and Y.-G. Guéhéneuc, "Is it a bug or an enhancement? a text-based approach to classify change requests", in *Conference of the center for advanced studies on collaborative research: meeting of minds*, 2008, pp. 304–318.