

From Inductive to Deductive: LLMs-Based Qualitative Data Analysis in Requirements Engineering

Syed Tauhid Ullah Shah¹, Mohamad Hussein¹, Ann Barcomb^{1,†} and Mohammad Moshirpour²

¹ University of Calgary, Calgary, Canada

² University of California, Irvine, Irvine, California, USA,

Abstract

Requirements Engineering (RE) is essential for developing complex and regulated software projects. Given the challenges in transforming stakeholder inputs into consistent software designs, Qualitative Data Analysis (QDA) provides a systematic approach to handling free-form data. However, traditional QDA methods are time-consuming and heavily reliant on manual effort. In this paper, we explore the use of Large Language Models (LLMs), including GPT-4, Mistral, and LLaMA-2, to improve QDA tasks in RE. Our study evaluates LLMs' performance in inductive (zero-shot) and deductive (one-shot, few-shot) annotation tasks, revealing that GPT-4 achieves substantial agreement with human analysts in deductive settings, with Cohen's Kappa scores exceeding 0.7, while zero-shot performance remains limited. Detailed, context-rich prompts significantly improve annotation accuracy and consistency, particularly in deductive scenarios, and GPT-4 demonstrates high reliability across repeated runs. These findings highlight the potential of LLMs to support QDA in RE by reducing manual effort while maintaining annotation quality. The structured labels automatically provide traceability of requirements and can be directly utilized as classes in domain models, facilitating systematic software design."

Keywords

Requirements Engineering, Qualitative Data Analysis, Large Language Models, Zero-shot Learning, Few-shot Learning, Natural Language Processing

1. Introduction

Requirements Engineering is a key process in developing large and complex software systems. It ensures that the software meets the needs of stakeholders by gathering, organizing, and managing their requirements systematically [1]. QDA is an emerging approach in RE that aids in analyzing unstructured data like interviews and surveys to identify patterns and insights [2, 3, 4]. One important step in QDA is labeling or coding, where pieces of text are categorized into themes to make the data more structured and meaningful [5]. This process helps improve traceability, consistency, and the quality of software design [6]. However, traditional QDA methods can be slow, inconsistent, and require a lot of manual work [7].

Recently, Large Language Models (LLMs), such as GPT-4 [8], Gemini [9], and LLaMA-2 [10], have shown great potential at processing and generating human-like text, making them useful for working with large sets of unstructured data. Unlike traditional models, LLMs use natural language prompts for tasks such as text classification [11], summarization [12], and translation [13]. Their adaptability across zero-shot and few-shot scenarios [14, 9] reduces reliance on extensive training data and computational

In: A. Hess, A. Susi, E. C. Groen, M. Ruiz, M. Abbas, F. B. Aydemir, M. Daneva, R. Guizzardi, J. Gulden, A. Herrmann, J. Horkoff, S. Kopczyńska, P. Mennig, M. Oriol Hilari, E. Paja, A. Perini, A. Rachmann, K. Schneider, L. Semini, P. Spoletini, A. Vogelsang. Joint Proceedings of REFSQ-2025 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track. Co-located with REFSQ 2025. Barcelona, Spain, April 7, 2025.

✉ syed.tauhidullahshah@ucalgary.ca (S. T. U. Shah); mohamad.hussein@ucalgary.ca (M. Hussein); ann.barcomb@ucalgary.ca (A. Barcomb); mmoshirp@uci.edu (M. Moshirpour)



© 2025 Copyright © 2025 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

resources. In RE, structured outputs like software specifications are essential, and LLMs can help by generating accurate and contextually relevant outputs [15].

In this study, we use LLMs, such as GPT-4, Mistral, and LLaMA-2, to assist in qualitative data annotation for RE, aiming to reduce manual effort and accelerate the analysis process. Our approach uses both inductive and deductive annotation. To facilitate the alignment of inductive and deductive with the NLP setup, we treated inductive annotation as zero-shot learning and used one-shot and few-shot learning for deductive annotation. Our experiments, conducted on two test cases (Library Management System and Smart Home System), demonstrate that our LLM-based approach achieved fair to substantial agreement with human analysts in deductive annotation tasks. Specifically, in both test cases, GPT-4 performed better than the other LLMs, showing stronger agreement with human analysts. Contextual examples in detailed prompts led to notable performance gains, especially during the shift from zero-shot to one-shot scenarios. Providing rich context was key, as it produced much better results than using limited or no context. Our findings demonstrate that LLMs can effectively support qualitative data annotation in RE, offering faster and more consistent results. Additionally, the structured labels generated by these models help create domain models, which are critical for systematic software design and development. This not only reduces manual effort but also ensures greater consistency and accuracy, improving the overall quality of software design.

Our work is structured around the following research key questions:

- **RQ1:** To what extent does our LLM-based approach align with human analysts in both inductive and deductive annotation tasks?
- **RQ2:** How do different prompt designs (zero-shot and few-shot) and lengths (short, medium, long) affect the accuracy and reliability of the annotations generated by LLMs?
- **RQ3:** How consistent are the LLM-generated labels across multiple runs?
- **RQ4:** How do various contextual settings affect the effectiveness of our LLM-based annotation approach?

Overall, our contributions can be summarized as follows:

- We conducted a comprehensive assessment of both open-source and proprietary LLMs to determine their utility in supporting QDA within RE. Our study spans various models, including GPT-4, Mistral, and LLaMA-2.
- We explored the effectiveness of different annotation strategies (inductive and deductive) across various settings (zero-shot, one-shot, and few-shot). Our findings illustrate the impacts of these strategies on the performance of LLMs, with deductive (few-shot) annotation achieving higher agreement with human analysts. For instance, GPT-4 reached a Cohen’s Kappa score of up to 0.738, indicating substantial agreement.
- We investigated the influence of prompt length and contextual information on the performance of LLMs. Detailed, context-rich prompts significantly enhanced the accuracy of LLMs. In the few-shot setting, the precision and recall for GPT-4 were notably high, at 0.80 and 0.79, respectively, demonstrating its effectiveness in closely mirroring human analytical processes.

2. Literature Review

In this literature review, we explore two critical areas: the role of QDA in RE (Section. 2.1) and the application of LLMs in RE (Section. 2.2) for QDA-assisted RE.

2.1. Qualitative Data Analysis (QDA)-based RE

QDA is a key technique in RE for analyzing unstructured stakeholder inputs, such as interviews and surveys, to extract patterns and generate actionable insights [16]. Qualitative labeling is used to

identify domain concepts and latent requirements. These coded insights are then mapped to classes or components in a domain model, ensuring that stakeholder needs are accurately reflected in the system design [17]. While QDA improves traceability and accuracy in requirements specification, traditional methods are labor-intensive, inconsistent, and prone to subjectivity [18, 19]. Tools like Computer Assisted Qualitative Data Analysis Software (CAQDAS) aim to support the process but often lack adaptability to dynamic RE environments [20]. Recent efforts like QDAcity-RE [20, 21] have shown that QDA techniques help extract domain concepts from unstructured stakeholder interviews and documentation. This approach uses manual qualitative coding to generate traceable domain models by mapping labeled requirements to classes or components, ensuring consistency and traceability in the design process. However, the repetitive and manual nature of these processes underscores the need for automation to improve scalability and efficiency.

2.2. Large Language Models (LLMs) in Requirements Engineering (RE)

LLMs, such as GPT-4, Mistral, and LLaMA-2, have shown promise in automating RE tasks like requirements classification, ambiguity detection, and documentation synthesis [22, 23]. Their adaptability across zero-shot and few-shot scenarios enables efficient processing of unstructured data with minimal training [14]. Recent studies have explored the application of LLMs in qualitative research within software engineering [24]. For example, Alhoshan et al. [25] demonstrated the potential of LLMs for requirements classification without task-specific training, while Kici et al. [26] showed the effectiveness of transfer learning for RE tasks. Despite this progress, applying LLMs to QDA for RE remains underexplored, presenting an opportunity to address limitations of traditional QDA and enhance scalability and accuracy in RE processes.

Although LLMs have been widely studied in RE and QDA independently, their integration for QDA in RE is still new. Using LLMs for QDA can greatly improve efficiency and accuracy by automating annotations and reducing errors from manual work, can simplify the process, make it more reliable and scalable, and better meet the changing demands of RE.

3. Qualitative Data Analysis (QDA)

For our study, we focused on two specific test cases: a Library Management System and a Smart Home system. The Library Management System test case involves managing resources like cataloging, user management, loans, and digital resources. The Smart Home System test case focuses on automating tasks such as security, energy control, and device management. While the two primary test cases were sourced from the PURE dataset [27], we supplemented these with additional SRS and FRS documents from the internet to ensure a comprehensive dataset. Following the extensive data collection, we applied QDA to our test cases. Our primary goal was to convert the requirement statements from these documents into actionable insights by assigning precise labels to distinct segments. These labels, akin to UML classes, help structure the requirements, making them more comprehensible and aiding their integration into the software development lifecycle. This structured approach ensures that the requirements are clear, precise, and aligned with the overall goals of the software engineering process.

To maintain precision and reliability, we assigned two independent analysts, C_1 and C_2 , to review and label the same set of requirement documents independently. Both analysts have a software engineering background, with C_1 having 1.5 years of experience and C_2 having 8 months of experience working with software requirements. First, both analysts (C_1 and C_2) labeled the requirement documents independently. We then measured their agreement using Cohen's Kappa¹. After that, they met to

¹Cohen's Kappa is a statistical measure used to assess the inter-rater agreement of qualitative (categorical) items. It considers the agreement occurring by chance and provides a more robust metric compared to simple percent agreement. A Kappa score of more than 0.70 typically indicates a substantial level of agreement between raters, reflecting a high degree of reliability in the annotation process.

discuss and resolve any differences, creating a unified set of labels. This iterative process combined their insights into a unified analytical framework. The total time and effort spent by the analysts in this QDA process are summarized in Table. 1. We reached a substantial agreement of 0.80 for the Library Management System and 0.78 for the Smart Home System. The Library Management System used labels such as 'Notification,' 'Loan,' 'Reservation,' 'Catalog,' etc., while the Smart Home System included 'Sensor,' 'Light,' 'Thermostat,' 'Device,' etc. These labels ensure stakeholder inputs are directly linked to corresponding elements in the domain model

Table 1

Time spent for the RE-QDA annotation process

Parameter	Amount/Duration
Analyst C_1	37 hours
Analyst C_2	29 hours
Meeting Duration	8 hours (12 meetings in total)
Entire Time Spent	74 hours

4. Methodology

4.1. Overview

Figure. 1 outlines our approach to integrating LLMs into QDA for RE. We begin by taking requirement statements (Section. 3) as input. The requirements are subsequently formatted into structured prompts optimized for inductive or deductive annotations (Section. 4.2). Inductive prompts, used in zero-shot learning, allow LLMs to identify patterns without predefined categories, while deductive prompts, supporting one-shot and few-shot learning, include examples for consistency with defined categories. LLMs (Section. 4.3) process these prompts to generate structured labels (Section. 4.4), which categorize and interpret requirements, providing actionable insights for further development. This approach simplifies the QDA process, reducing manual effort while leveraging LLM capabilities effectively.

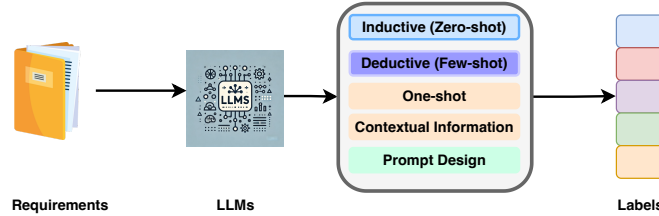


Figure 1: Overview of the methodology integrating LLMs into QDA for RE. The process includes collecting requirement statements, designing prompts, feeding them to LLMs, and generating output labels.

4.2. Prompt Design

We created clear and structured prompts to convert the collected requirements into a format that LLMs can understand and label. Table 2 summarizes our prompt templates, while Table 3 provides details on our context levels. Our design considers three independent factors:

1. Shot Type: This factor refers to the number of examples included in the prompt. In a *zero-shot* prompt, no examples are provided, so the LLM relies entirely on its built-in knowledge. A *one-shot* prompt includes one example to guide the model, while a *few-shot* prompt provides several examples to clearly show the desired labeling.

2. Prompt Length: This factor measures how much instruction is given. A *short* prompt provides minimal instructions, a *medium* prompt adds additional details, and a *long* prompt gives in-depth guidance. For instance, a long prompt might explain specific aspects of QDA such as traceability, stakeholder intent, and consistency.

Table 2
Summary of Prompt Templates by Shot Type, Length, and Contextuality

Category	Prompt Type	Prompt Description (Example Template)
<i>Zero-shot Prompts (Inductive)</i>	Short	"Analyze this {requirement} and respond with ONLY a single category label (1 word)."
	Medium	"You are analyzing software requirements for the {system_type} system. Respond with ONLY a single category label that best captures the main functionality for the following {requirement}."
	Long	"You are performing Qualitative Data Analysis on requirements for a {system_type} system. {context}. Analyze the requirement below and respond with ONLY a single categorical label (1 word) that best represents its main functionality for the following {requirement}."
<i>Few-shot Prompts (Deductive)</i>	Short	"Analyze requirements and respond with ONLY a single Qualitative Data Analysis-based label. Examples: {example1}, {example2}, {example3}."
	Medium	"For a {system_type} system, respond with ONLY a single Qualitative Data Analysis-based label that best represents the functionality. Examples: {example1}, {example2}, {example3}."
	Long	"You are performing Qualitative Data Analysis on requirements for a {system_type} system. {context} Given the following examples: Example 1: {example1} (Label: {label1}) Example 2: {example2} (Label: {label2}) Example 3: {example3} (Label: {label3}) Analyze the requirement below and respond with ONLY a single Qualitative Data Analysis-based label (1 word) that represents its main functionality."

3. Contextual vs. Non-Contextual: This aspect determines whether the prompt includes background information. Non-contextual prompts provide only the requirement statement, while contextual prompts offer system details to improve understanding. We define three levels: no context (requirement only), some context (brief system description), and full context (comprehensive system details).

4.3. Model Selection

Table 3
Context Levels for Prompt Design

Context Level	Description and Example
No Context	Only the requirement is provided. Example: "Requirement: {requirement}"
Some Context	A brief system description is added. Example: "This is a Library Management System that handles cataloging, user management, and loans. Requirement: {requirement}"
Full Context	A comprehensive system description is provided, detailing functionalities and design specifics. Example: "The Library Management System (LMS) manages all aspects of a modern library, including resource cataloging, loan processing, digital resource management, and administrative reporting. Requirement: {requirement}"

We used state-of-the-art LLMs, including GPT-4 [8], Mistral [28], and LLaMA-2 [10], for their abilities in understanding and generating natural language and suitability for the complex task of QDA in RE. We prompt these models with specific software requirement data to understand the context of requirements, recognize domain-specific terminology, and map requirement statements to relevant labels.

Table 4

Comparison of Cohen’s Kappa Scores for Different Models Across Both Test Cases

Test Case	Setting	Llama 2	Mistral	GPT-4
Library Management System	Zero-shot	0.516	0.526	0.543
	One-shot	0.675	0.685	0.690
	Few-shot	0.730	0.734	0.738
Smart Home System	Zero-shot	0.514	0.530	0.541
	One-shot	0.681	0.686	0.689
	Few-shot	0.728	0.730	0.734

4.4. Output Labels

Our approach focuses on generating labels that organize and interpret requirement statements, converting unstructured data into clear and actionable insights. These labels are critical for understanding stakeholder needs and ensuring that requirements align with their expectations [29]. By improving communication among teams, the labels also play a key role in creating domain models, which are essential for systematic software design [21]. To achieve accurate and relevant labels, we employ both inductive and deductive strategies, supported by contextual prompts. This dual strategy improves the precision and relevance of the labeling process. Additionally, these QDA-based annotations ensure automatic traceability by linking each label back to its corresponding stakeholder input [20].

5. Results

5.1. Evaluation Metrics

We assessed the performance of the LLMs using several key metrics to evaluate their accuracy and agreement in annotation tasks. Inter-rater agreement was measured using Cohen’s Kappa, which quantifies the level of agreement between the labels generated by the LLMs and those assigned by human analysts, with higher values indicating stronger agreement. To evaluate the consistency of the labels across multiple experimental runs, we analyzed the standard deviation (SD) and the Intraclass Correlation Coefficient (ICC). A lower SD indicates minimal variability in the labels, while ICC values above 0.85 demonstrate excellent reliability. In addition to reliability and consistency, we evaluated the accuracy of the LLMs, which measures the proportion of correct labels out of all predictions. Precision was used to determine how many of the labels identified by the LLMs were correct, providing insights into their ability to avoid false positives. Recall assessed the ability of LLMs in the identification of all relevant labels, minimizing the risk of missing important instances (false negatives). Finally, we used F1-score, the harmonic mean of precision and recall, to provide a balanced measure of the performance of the models, with higher scores indicating a good trade-off between precision and recall. In this study, we used only the labels on which both analysts reached consensus as the ground truth for evaluating LLM performance.

5.2. Implementation

We carried out all experiments using Python and PyTorch². For Mistral and LLaMA-2 models, we used the 7B configuration from the Hugging Face’s Transformers library³, which provides access to pre-trained models while for GPT-4, we used the GPT-4 Turbo API⁴. To ensure fair comparisons, we set the temperature parameter to 0.0 across all models, which minimizes randomness and makes outputs

²<https://pytorch.org/>

³<https://huggingface.co/transformers/>

⁴<https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4>

Table 5

Comparison of Cohen’s Kappa Scores for Models with different Prompt Lengths (Short, Medium, Long)

Model	Library Management			Smart Home		
	Short	Medium	Long	Short	Medium	Long
Llama 2	0.629	0.686	0.707	0.624	0.698	0.705
Mistral	0.645	0.699	0.713	0.633	0.685	0.712
GPT-4	0.641	0.691	0.738	0.631	0.681	0.734

consistent. The experiments were conducted on high-performance computing clusters equipped with NVIDIA A100 GPUs to handle the computational demands. The source code for all experiments and evaluations is publicly available⁵.

5.3. LLMs vs. Human Analysts (RQ1)

To evaluate the effectiveness of LLMs in aiding QDA-based annotation tasks within RE, we compared their performance against human analysts for both inductive and deductive settings. We used Cohen’s Kappa, a widely recognized statistical measure for assessing inter-rater agreement, to quantify agreement levels between LLM-generated labels and those derived by human analysts (described in detail in Section. 3). This measure highlights the reliability and consistency of the LLM’s performance in replicating human judgment, aligning with practices in qualitative research [30] and LLM-assisted content analysis [31].

Table. 4 reports the Cohen’s Kappa results for various prompt designs (zero-shot, one-shot, few-shot) and test cases (Library Management System and Smart Home System). Our empirical assessment across various settings for both test-cases yielded significant insights into the capabilities of LLMs. Notably, GPT-4 consistently outperformed other models such as LLaMA-2 and Mistral, achieving the highest Cohen’s Kappa scores. Specifically, in the few-shot setting, GPT-4 achieved scores of 0.738 and 0.734 for the Library Management System and the Smart Home System, respectively, indicating substantial agreement with human analysts and highlighting its robustness in these settings.

However, it is important to note that the agreement levels in the zero-shot setting were around 0.54, which is not typically considered a strong outcome. This observation suggests that while LLMs can approach the performance of human analysts in scenarios where some guidance (one-shot or few-shot) is provided, their effectiveness in fully autonomous, inductive annotation tasks (zero-shot) remains limited. This analysis highlights that, although LLMs show promise, particularly in deductive settings where they can match or even exceed human performance, they still require refinement for inductive tasks where no initial guidance is given. This detailed understanding addresses **RQ1**, indicating that while LLMs hold significant potential to support human efforts in RE annotation processes, their current application is more reliable in deductive annotation tasks than inductive ones.

5.4. Influence of Prompt Design on Annotation Outcomes (RQ2)

To assess the impact of different prompt lengths, we executed a series of experiments across the two distinct test cases. The results, summarized in Table. 5, indicate that while long prompts generally provide the best performance, medium prompts also offer a good balance of context and efficiency. Short prompts, although less detail-intensive, often fall short in tasks requiring detailed contextual understanding.

This analysis directly addresses **RQ2**, demonstrating that careful prompt design is essential for maximizing the effectiveness of LLMs in annotation tasks within RE. Also, the finding is consistent with the

⁵<https://github.com/SyedTauhidUllahShah/LLM4QDARE>

Table 6

Consistency Analysis of LLM-Generated Labels Across Multiple Runs for Both Test Cases

Test Case	Metric	Llama 2	Mistral	GPT-4
Library Management	SD	0.057	0.048	0.034
	ICC	0.87	0.89	0.93
Smart Home	SD	0.062	0.051	0.037
	ICC	0.85	0.88	0.92

broader literature [32, 33], which emphasizes that the detailed contextual information in long prompts significantly enhances LLM performance by reducing ambiguity. Our findings highlight the potential for optimizing LLM performance in practical applications by tailoring prompts to balance context and efficiency.

5.5. Consistency Analysis of LLM-Generated Labels Across Multiple Runs (RQ3)

The consistency analysis of LLM-generated labels across multiple runs, as shown in Table. 6, revealed that GPT-4 exhibited the highest consistency among the tested models. Specifically, GPT-4 achieved the lowest standard deviations of 0.034 for the Library Management System and 0.037 for the Smart Home System. Additionally, GPT-4 obtained the highest ICC values of 0.93 and 0.92 for the Library Management System and Smart Home System, respectively. These results indicate a high degree of reliability and stability in the generated labels, surpassing the performance of LLaMA-2 and Mistral, which also demonstrated good consistency but with slightly higher variability.

The high ICC values (>0.85) across all models affirm that LLM-generated labels are consistently reproducible within the same class, ensuring reliable outputs that closely align with the performance of human analysts. These findings show that GPT-4 is a reliable tool for helping with QDA in RE, making it easier to extract and organize insights from requirements data with less manual work.

5.6. Impact of Contextual Backgrounds (RQ4)

To address **RQ4**, we evaluated the impact of varying levels of contextual backgrounds on the effectiveness of LLM-generated labels. Specifically, we adjusted the amount of context provided in the prompts, ranging from no context to full context. The results, as shown in Table. 7, demonstrated that the inclusion of richer contextual information in the prompts significantly improved the performance of all evaluated models, including LLaMA-2, Mistral, and GPT-4.

Specifically, GPT-4 exhibited the highest Cohen’s Kappa scores across all scenarios, achieving scores of 0.738 for the Library Management System and 0.734 for the Smart Home System in the full context setting. These findings indicate that GPT-4 is particularly effective at leveraging detailed contextual information to generate accurate and consistent labels.

The improvement in performance with increased context suggests that providing comprehensive background information enables LLMs to better understand and interpret the requirements, resulting in more precise annotation. This highlights the importance of designing context-rich prompts to maximize the potential of LLMs for automating and refining QDA processes within RE. By incorporating detailed contextual information, LLMs can deliver outputs that accurately reflect the complexities of the requirements, thereby improving the accuracy and reliability of the annotation process.

Table 7

Cohen’s Kappa Analysis of Contextual Information Across different Levels of Contextual Information

Test Case	Context	Llama 2	Mistral	GPT-4
Library Management	No Context	0.663	0.674	0.712
	Some Context	0.682	0.689	0.718
	Full Context	0.707	0.713	0.738
Smart Home	No Context	0.673	0.682	0.713
	Some Context	0.691	0.701	0.722
	Full Context	0.705	0.712	0.734

Table 8

Detailed Performance Metrics for Different Models in Zero-shot, One-shot, and Few-shot Settings Across Test Cases

Setting	Model	Library Management				Smart Home			
		Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
Zero-shot	Llama 2	0.68	0.65	0.64	0.645	0.67	0.64	0.63	0.635
	Mistral	0.70	0.68	0.67	0.675	0.69	0.66	0.65	0.655
	GPT-4	0.72	0.70	0.69	0.695	0.71	0.68	0.67	0.675
One-shot	Llama 2	0.78	0.72	0.71	0.715	0.77	0.71	0.70	0.705
	Mistral	0.80	0.74	0.73	0.735	0.79	0.73	0.72	0.725
	GPT-4	0.82	0.76	0.75	0.755	0.81	0.75	0.74	0.745
Few-shot	Llama 2	0.84	0.76	0.74	0.750	0.83	0.75	0.73	0.740
	Mistral	0.85	0.78	0.76	0.770	0.84	0.77	0.75	0.760
	GPT-4	0.86	0.80	0.79	0.795	0.85	0.79	0.78	0.785

5.7. Performance Evaluation with Detailed Metrics

To further validate our results, we incorporated additional evaluation metrics: accuracy, precision, recall, and F1-score. The detailed performance evaluation, presented in Table. 8, shows that GPT-4 consistently outperforms LLaMA-2 and Mistral across all metrics. Specifically, GPT-4 achieves the highest accuracy, precision, and recall in both zero-shot and few-shot settings for the Library Management and Smart Home test cases. Although in the inductive scenario, the model is not provided with explicit examples, it still outputs a single label per requirement that is evaluated against the ground truth. In the deductive scenario, implemented as few-shot learning, the model is guided by explicit examples to generate labels. In both cases, the task is treated as a multi-class classification problem. For instance, in the few-shot setting for the Library Management test case, GPT-4 achieves an accuracy of 0.86, a precision of 0.80, recall of 0.79, and an F1-score of 0.79, demonstrating its superior ability to correctly and consistently categorize requirement statements.

Similarly, in the Smart Home test case, GPT-4 again leads with an accuracy of 0.85, a precision of 0.79, a recall of 0.78, and an F1-score of 0.785 in the few-shot setting. This analysis supports our earlier findings from Cohen’s Kappa and ICC, showing that GPT-4 is reliable for automating QDA tasks in RE. The higher precision and recall suggest that GPT-4 not only identifies the correct labels more often but also misses fewer important instances, making the annotations more complete and accurate.

6. Threats to Validity

In this section, we discuss the potential threats to the validity of our study on the application of LLMs for QDA in RE.

6.1. Internal Validity

One challenge in this study is the potential bias in pre-trained LLMs such as GPT-4, Mistral, and LLaMA-2. Since these models are trained on vast datasets, their outputs may reflect underlying biases that could skew the annotation results and fail to fully capture the nuances of RE. To minimize this risk, we carefully designed prompts with detailed context to guide the models toward more accurate and relevant annotations. Another concern is the consistency of human annotations. Different analysts may interpret and label the same requirement statements in slightly different ways, which could introduce inconsistencies in the dataset used for evaluation. To address this, we used an inter-rater reliability phase, where analysts reviewed their annotations together, resolving discrepancies to improve label consistency. Prompt design also plays a crucial role in the accuracy of LLM-generated annotations. Poorly structured or vague prompts can lead to unreliable results. To improve performance, we tested prompts with different lengths and levels of contextual information, refining them through an iterative process to ensure clarity and effectiveness.

6.2. External Validity

Our study evaluates LLM performance using two test cases, Library Management and Smart Home systems, which may not fully capture the diversity of software systems in practice. Results could vary when applied to different domains, particularly those with unique complexities or highly specialized requirements. The dataset, while sourced from multiple documents, may not represent the full range of real-world projects. A broader selection of requirement documents covering various industries and project types would strengthen the evaluation and improve the generalizability of our findings. Contextual information in prompts also plays a key role in guiding LLMs toward accurate annotations, but our prompts may not fully capture every detail of different RE contexts. Ensuring clarity and relevance across diverse scenarios remains a challenge. Further refinement, incorporating real-world feedback, is needed to enhance the applicability of this approach.

7. Conclusion and Future Work

This paper explored the application of LLMs, specifically LLM, Mistral, and LLaMA-2, to aid and enhance the annotation processes in RE. Our findings demonstrate that GPT-4, in particular, significantly reduces the manual effort required for annotation, achieving high levels of accuracy and consistency comparable to human analysts. The performance of these models is notably improved with detailed, context-rich prompts, underscoring the importance of prompt design in leveraging LLM capabilities effectively. Our work highlights that while GPT-4 and other LLMs show promise in deductive annotation tasks (one-shot and few-shot settings), achieving substantial agreement with human analysts, their effectiveness in inductive annotation tasks (zero-shot) remains limited. This calls for further development and optimization of LLM strategies to enhance their performance across all types of annotation tasks. The potential for broader adoption of LLMs in RE is clear, suggesting that these models can aid QDA, increase efficiency, and reduce subjectivity. The structured labels generated by LLMs not only improve the efficiency and reliability of the QDA process but also facilitate the creation of domain models, simplifying the software design process and enhancing overall project efficiency. Future work should focus on extending these results to more diverse scenarios and further refining the training processes to address any inherent model biases. By doing so, the utility and reliability of LLMs in enhancing various aspects of software development processes can be significantly expanded.

References

- [1] B. H. Cheng, J. M. Atlee, Research directions in requirements engineering, *Future of software engineering (FOSE'07)* (2007) 285–303.

- [2] D. Carrizo, O. Dieste, N. Juristo, Systematizing requirements elicitation technique selection, *Information and Software Technology* 56 (2014) 644–669.
- [3] J. Mucha, The QDAcity-RE-RS Method for Creating Complete, Consistent, and Traceable Requirements Specifications, Friedrich-Alexander-Universitaet Erlangen-Nuernberg (Germany), 2023.
- [4] A. Kaufmann, J. Krause, N. Harutyunyan, A. Barcomb, D. Riehle, A validation of QDAcity-RE for domain modeling using qualitative data analysis, *Requirements Engineering* (2021). URL: <https://link.springer.com/article/10.1007/s00766-021-00360-6>. doi:<https://doi.org/10.1007/s00766-021-00360-6>.
- [5] J. Saldaña, *The coding manual for qualitative researchers* (2021).
- [6] C. Treude, Qualitative data analysis in software engineering: Techniques and teaching insights, *arXiv preprint arXiv:2406.08228* (2024).
- [7] S. Tsang, An experiment exploring the theoretical and methodological challenges in developing a semi-automated approach to analysis of small-n qualitative data, *arXiv preprint arXiv:2002.04513* (2020).
- [8] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, *arXiv preprint arXiv:2303.08774* (2023).
- [9] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al., Gemini: a family of highly capable multimodal models, *arXiv preprint arXiv:2312.11805* (2023).
- [10] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, *arXiv preprint arXiv:2307.09288* (2023).
- [11] X. Sun, X. Li, J. Li, F. Wu, S. Guo, T. Zhang, G. Wang, Text classification via large language models, in: *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.
- [12] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, T. B. Hashimoto, Benchmarking large language models for news summarization, *Transactions of the Association for Computational Linguistics* 12 (2024) 39–57.
- [13] B. Zhang, B. Haddow, A. Birch, Prompting large language model for machine translation: A case study, in: *International Conference on Machine Learning*, PMLR, 2023, pp. 41092–41110.
- [14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, *Advances in neural information processing systems* 33 (2020) 1877–1901.
- [15] M. Krishna, B. Gaur, A. Verma, P. Jalote, Using llms in software requirements specifications: An empirical evaluation, *arXiv preprint arXiv:2404.17842* (2024).
- [16] B. Nuseibeh, S. Easterbrook, Requirements engineering: a roadmap, in: *Proceedings of the Conference on the Future of Software Engineering*, 2000, pp. 35–46.
- [17] A. Kaufmann, J. Krause, N. Harutyunyan, A. Barcomb, D. Riehle, A validation of qdacity-re for domain modeling using qualitative data analysis, *Requirements Engineering* 27 (2022) 31–51.
- [18] N.-C. Chen, R. Kocielnik, M. Drouhard, V. Peña-Araya, J. Suh, K. Cen, X. Zheng, C. R. Aragon, Challenges of applying machine learning to qualitative coding, in: *ACM SIGCHI Workshop on Human-Centered Machine Learning*, 2016.
- [19] B. Glaser, A. Strauss, *Discovery of grounded theory: Strategies for qualitative research*, Routledge, 2017.
- [20] A. Kaufmann, D. Riehle, The QDAcity-RE method for structural domain modeling using qualitative data analysis, *Requirements Engineering* 24 (2019) 85–102.
- [21] A. Kaufmann, A. Barcomb, D. Riehle, Supporting interview analysis with autocoding, in: *53rd Hawaii International Conference on System Sciences, HICSS 2020, Maui, Hawaii, USA, January 7-10, 2020, ScholarSpace*, 2020, pp. 1–10.
- [22] A. Vogelsang, J. Fischbach, Using large language models for natural language processing tasks in requirements engineering: A systematic guideline, *arXiv e-prints* (2024) arXiv-2402.
- [23] A. Fan, B. Gokkaya, M. Harman, M. Lyubarskiy, S. Sengupta, S. Yoo, J. M. Zhang, Large language models for software engineering: Survey and open problems, *arXiv preprint arXiv:2310.03533*

(2023).

- [24] M. Bano, R. Hoda, D. Zowghi, C. Treude, Large language models for qualitative research in software engineering: exploring opportunities and challenges, *Automated Software Engineering* 31 (2024) 8.
- [25] W. Alhoshan, A. Ferrari, L. Zhao, Zero-shot learning for requirements classification: An exploratory study, *Information and Software Technology* 159 (2023) 107202.
- [26] D. Kici, G. Malik, M. Cevik, D. Parikh, A. Basar, A bert-based transfer learning approach to text classification on software requirements specifications., in: *Canadian AI*, 2021.
- [27] A. Ferrari, G. O. Spagnolo, S. Gnesi, Pure: a dataset of public requirements documents, *Unspecified Journal Unspecified Volume* (2023) Unspecified Pages.
- [28] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al., Mistral 7b, *arXiv preprint arXiv:2310.06825* (2023).
- [29] K. E. Wieggers, J. Beatty, *Software requirements*, Pearson Education, 2013.
- [30] M. L. Coleman, M. Ragan, T. Dari, Intercoder reliability for use in qualitative research and evaluation, *Measurement and Evaluation in Counseling and Development* 57 (2024) 136–146.
- [31] R. Chew, J. Bollenbacher, M. Wenger, J. Speer, A. Kim, Llm-assisted content analysis: Using large language models to support deductive coding, *arXiv preprint arXiv:2306.14924* (2023).
- [32] M. Turpin, J. Michael, E. Perez, S. Bowman, Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting, *Advances in Neural Information Processing Systems* 36 (2024).
- [33] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, *Advances in neural information processing systems* 35 (2022) 24824–24837.