DOI:10.5937/IIZS26263R

APPLICATION OF LARGE LANGUAGE MODELS IN REQUIREMENTS ENGINEERING

Review paper

Branislava Radu, Ljubica Kazi, Ivana Berković, Zeljko Stojanov
University of Novi Sad, Technical Faculty "Mihajlo Pupin", Zrenjanin, Serbia

Abstract: Requirements engineering plays a significant role in software development, while the development of large language models (LLMs) provides new opportunities to support and automate various activities within this process. The aim of this paper is to analyze the application of LLMs in requirements engineering, with particular focus on the tasks in which they are applied, the methods used to evaluate their outputs and the limitations of their use. Based on a review of selected literature, the applications of LLMs in requirements elicitation, specification, analysis and validation were analyzed, along with different models, prompt formulation approaches and criteria for assessing the quality of generated outputs. The analysis indicates that LLMs can contribute to the automation of certain activities, improve requirements quality and reduce the time required for their processing, while the obtained results depend on the choice of model, task characteristics, available context and prompt formulation. Key challenges include the generation of inaccurate and biased content, context-related issues, ambiguous requirements, data protection and security concerns. Further development should focus on more reliable evaluation, improved context management and more effective collaboration between LLMs and experts.
Key words: Requirements engineering, large language models, LLMs, generative artificial intelligence, evaluation

INTRODUCTION

Requirements Engineering (RE) represents one of the key phases of software development. The complexity of these activities, as well as the time and resources required to perform them, has encouraged the exploration of possibilities for their automation (Ronanki et al., 2024). The successful implementation of software projects largely depends on the quality of defined requirements, as clearly and unambiguously formulated requirements provide the foundation for the further development of a software system (Lubos et al., 2024).

The development of generative artificial intelligence, particularly large language models (LLMs), has opened new possibilities for automating and supporting Requirements Engineering activities. Owing to their ability to process natural language, LLMs are applied to various tasks, including the elicitation, analysis, specification and validation of software requirements (Arora et al., 2024). Although their application offers considerable potential for improving these activities, it is also accompanied by certain limitations. These include the possibility of generating inaccurate, fabricated and biased content, limited explainability, difficulties in handling ambiguous and complex requirements, as well as challenges related to data protection and security. Excessive reliance on these models may also lead to the neglect of the specific context and needs of stakeholders, making human verification and supervision of the obtained results necessary (Marques et al., 2024).

The aim of this paper is to analyze the application of large language models in Requirements Engineering, focusing on the tasks in which they are applied, the models used and the ways in which they are applied, approaches to evaluating their results and the limitations that arise across different activities. The paper examines various applications of LLMs in Requirements Engineering, as well as the influence of prompt formulation and model selection on the quality of the obtained results. Based on a literature review, specific models, tasks and evaluation criteria are identified with the aim of examining their capabilities and limitations, as well as identifying underexplored areas and potential directions for future research.

The paper is organized as follows. Following the introduction, the basic concepts of Requirements Engineering and large language models are presented. The applications of

[1]Corresponding author, branislava.radu@tfzr.rs, ORCID: 0009-0004-3337-8445
[2]ljubica.kazi@tfzr.rs, ORCID: 0000-0001-8089-3029
[3]ivana.berkovic@tfzr.rs, ORCID: 0000-0003-2366-3942
[4]zeljko.stojanov@uns.ac.rs, ORCID: 0000-0001-6930-5337

LLMs in Requirements Engineering are then discussed, including their use across different phases of the process and in specific tasks. This section also analyzes selected models, their input and output data and the influence of prompt formulation on performance. Furthermore, approaches to evaluating results, limitations of their application and directions for future research are discussed. Finally, the conclusions and the main findings of the research are presented.

BACKGROUND

Requirements Engineering

Requirements Engineering is a field of software engineering focused on the systematic identification, documentation, analysis and verification of system and software requirements throughout their life cycle. The process involves several interconnected activities. During requirements elicitation, information about stakeholders' needs is collected, while requirements analysis involves their assessment and the identification of shortcomings such as ambiguity, inconsistency and incompleteness. Modeling enables requirements to be represented in a structured form, while requirements management involves tracking changes and maintaining their relationships with other development artifacts. Validation and verification are used to determine whether the requirements correspond to the needs and expectations of stakeholders and whether the system specification satisfies the defined requirements. These activities are primarily based on natural language and include tasks such as classification, traceability and defect detection. A key artifact is the System/Software Requirements Specification (SRS), which defines the system's functionalities, quality attributes and constraints and serves as a basis for design, implementation and testing. In addition to the System/Software Requirements Specification, modern Requirements Engineering also uses other sources, such as user stories, user feedback and legal and regulatory documents. This increases the scope of available information but also the co (Zadenoori et al., 2025)

Large Language Models

Large Language Models (LLMs) are artificial intelligence systems that enable the processing and generation of natural language text. Their training is based on large collections of textual data, from which the models learn language patterns, meanings and relationships between words in different contexts. Most modern LLMs are based on the Transformer architecture, which uses a self-attention mechanism to efficiently process context. This structure enables the application of models to tasks such as text generation and summarization, translation and question answering, with the Generative Pre-trained Transformer (GPT) being one of the most well-known examples. Although LLMs can be used in various domains, their reliability is not always guaranteed, as they may generate inaccurate information (Zadenoori et al., 2025)

APPLICATION OF LLMs IN REQUIREMENTS ENGINEERING

Application of LLMs Across the Phases of Requirements Engineering

This section presents large language models and their application in various Requirements Engineering activities, with an overview of the tasks in which they are applied, as well as the types of input and output data.

Large language models are increasingly being applied to various Requirements Engineering activities, including requirements extraction, classification, analysis and generation, as well as tasks that connect Requirements Engineering with other software engineering activities. Large Language Models for Requirements Engineering (LLM4RE) represents a relatively new research direction that has developed more intensively since 2023, with the application

of language models expanding from predominantly analytical tasks to more complex generative activities. (Zadenoori et al., 2025)

The application of LLMs encompasses different phases of Requirements Engineering, from elicitation to validation. During elicitation, LLMs can analyze documentation, identify missing information, formulate questions and suggest requirements, as well as support virtual interviews with stakeholders. During specification, they structure information and generate user stories, functional requirements and non-functional requirements. During analysis, they assess requirements quality and identify ambiguities, incompleteness, inconsistencies and potential risks, as well as suggest additional requirements and changes. During validation, they support requirements verification, simulate different stakeholder perspectives and generate acceptance criteria and test scenarios (Arora et al., 2024).

These applications encompass numerous specific tasks, ranging from completing incomplete requirements, assessing quality and detecting ambiguities to modeling, formalization and the generation of code and test artifacts. (Hemmat et al., 2025). Prompt-based LLMs have been investigated for the automatic classification of software requirements (Binkhonain & Alfayez, 2025). For example, Bidirectional Encoder Representations from Transformers was used to complete missing parts of requirements, while Chat Generative Pre-trained Transformer (ChatGPT) and Generative Pre-trained Transformer 3 (GPT-3.5) were applied to assess quality and formulate questions for requirements elicitation (Jin et al., 2024). GPT models are primarily used for generative and analytical tasks, Bidirectional Encoder Representations from Transformers for classification and analysis, Text-to-Text Transfer Transformer and Large Language Model Meta AI 2 (Llama 2) for reformulation and summarization, while multimodal models are used to connect textual and visual information (Hemmat et al., 2025).

Depending on the task, LLMs process different types of input data and generate different outputs. The most common inputs include SRS documents, user stories, technical documentation, models and code, while outputs include structured requirements, classifications, Unified Modeling Language (UML) diagrams, formal specifications and code (Hemmat et al., 2025). Recent research has extended these applications to LLM agents and multi-agent systems for more complex activities, such as document generation, user story analysis, security requirements formulation and regulatory compliance checking (Jin et al., 2024).

Applications of LLMs in Specific Requirements Engineering Tasks

This section presents specific applications of LLMs in various Requirements Engineering tasks, as well as the influence of model selection and prompt formulation on the obtained results.

One of the tasks in which the application of LLMs has been investigated is software requirements elicitation. In one study, GPT-3.5, was used to generate questions for surveys and interviews based on short and simple prompts. The quality of the questions was evaluated according to readability, clarity, relevance and completeness. For readability, the Flesch Reading Ease, which determines how easy a text is to read and the Flesch-Kincaid Grade Level, which estimates text complexity, were used. The average Flesch score was approximately 40, while the questions were generally assessed as clear and complete. Lower readability was observed in domains with more specialized terminology, whereas questions related to clothing retail were the simplest. The findings indicate that GPT-3.5 can support initial elicitation activities, while more precise prompt formulation can contribute to more focused and specific results (Yeow et al., 2024).

LLMs can be applied to generate Software Requirements Specification (SRS) documents. In one study, Generative Pre-trained Transformer 4 (GPT-4) and CodeLlama-34b were used to generate an SRS for a university student club management portal, and their outputs were compared with a human-authored SRS. The quality of individual requirements and the overall document was evaluated. At the level of individual requirements, clarity, understandability, correctness and verifiability were assessed, while at the level of the complete SRS, internal consistency, non-redundancy, completeness and conciseness were

considered. The human-authored SRS achieved the highest overall score, while CodeLlama-34b approached or exceeded the human result in some categories. CodeLlama-34b generated a shorter document containing a larger number of requirements in most sections; however, irrelevant, generic and incorrectly classified requirements were identified, as well as cases in which multiple requirements were combined into a single requirement. These issues could negatively affect understandability and lead to ambiguity. A more detailed analysis of individual characteristics showed that the human-authored SRS achieved better results in terms of clarity and understandability, whereas the LLMs performed better in terms of correctness. ChatGPT generated fewer but more precise performance requirements, while CodeLlama-34b provided more detailed security requirements. For design constraints, the human-authored SRS received the highest evaluation, whereas CodeLlama-34b showed difficulties in transferring requirements from other sections and including irrelevant requirements. For external interfaces, both models performed worse than the human-authored SRS. During subsequent correction, ChatGPT corrected 10 out of 11 problematic requirements, while CodeLlama-34b corrected four. At the same time, the LLMs significantly reduced the time required to create the SRS. CodeLlama-34b generated more detailed, complete and consistent documents, whereas documents generated using ChatGPT were more concise and contained less repetition. During validation, ChatGPT showed smaller average deviations from the evaluations provided by human evaluators. The findings indicate the potential of LLMs for more efficient specification generation, but also reveal limitations in contextual understanding and the correct classification of requirements, making human validation necessary (Krishna et al., 2024).

LLMs are also applied to the analysis and quality assessment of software requirements. One study investigated Llama 2 with 70 billion parameters for defect detection, justification of assessments and the suggestion of improved requirement formulations. The quality of requirements was evaluated according to characteristics defined by the ISO/IEC/IEEE 29148 standard: appropriate level of abstraction, completeness, consistency, correctness, feasibility, necessity, uniqueness, unambiguity and verifiability. The results were compared with assessments provided by experienced experts. The model achieved high recall but lower precision due to the identification of problems that did not actually exist, making human verification necessary. The explanations were generally assessed as convincing, while the proposed reformulations were considered improvements. The findings indicate that LLMs can provide useful support for reviewing and improving requirements quality, but they cannot replace human judgment (Lubos et al., 2024).

In addition to model selection, the results of LLM4RE tasks are influenced by prompt formulation and temperature settings, where lower temperature values produce more stable responses, while higher values generate more diverse responses. One study examined five prompting patterns for Generative Pre-trained Transformer 3.5 Turbo (GPT-3.5-turbo) on tasks involving the classification of functional and non-functional requirements and the identification of relationships between requirements. The "Question Refinement" pattern achieved the best results for requirements traceability, while "Cognitive Verifier" and "Question Refinement" were the most successful for binary classification. The findings confirm that the choice of prompting pattern and temperature significantly affects model performance. Precision, recall, F1-score and accuracy were used to evaluate performance (Ronanki et al., 2024). In addition, model adaptation techniques such as fine-tuning, prompt and context tuning and Low-Rank Adaptation are being investigated; however, most studies still rely on pretrained models without additional tuning (Hemmat et al., 2025).

Evaluation and Limitations of LLM Applications in Requirements Engineering

This section discusses evaluation methods, limitations of LLM applications and directions for future research.

The evaluation of LLM outputs in Requirements Engineering involves quantitative, qualitative and mixed-method approaches. In the analyzed studies, quantitative and mixed-method approaches were most frequently applied, while qualitative approaches were less

represented. Quantitative evaluation most commonly uses metrics such as precision and recall, whereas qualitative evaluation is based on thematic analysis. The mixed-method approach is particularly suitable for generative tasks, such as modeling and requirements elicitation, because a single reference solution often does not exist for such tasks. Instead, multiple generated solutions may satisfy the defined criteria, making expert assessment useful for evaluating their quality, contextual appropriateness and completeness, while involving multiple evaluators can contribute to a more objective assessment. (Zadenoori et al., 2025)

Depending on the task, different metrics are used to assess the quality and reliability of the results. Correctness and completeness evaluate the accuracy and coverage of the results, while precision, recall and F1-score measure the model's performance in identifying relevant elements. Agreement and consistency metrics assess the alignment of results with human judgments or standards, as well as the stability of outputs. These measures are complemented by qualitative and expert assessments used to evaluate the relevance, clarity, understandability and usability of generated results (Hemmat et al., 2025).

Although LLMs demonstrate considerable potential for application, their use is limited by inaccurate, fabricated and biased content, limited explainability, difficulties in handling ambiguous and complex requirements, as well as data protection and security concerns. Excessive reliance on these models may lead to the neglect of context and stakeholder needs; therefore, human verification and supervision are necessary (Marques et al., 2024). Particular difficulties arise when dealing with domain-specific, ambiguous and conflicting requirements, as well as when processing large documents, where context loss and inconsistencies may occur (Arora et al., 2024).

The use of GenAI in software engineering raises concerns related to dependability, accuracy, data accessibility and transparency, while requirements engineering remains an area that requires further research (Nguyen-Duc et al., 2025).Limitations have also been identified in specific experimental studies. The reliability of results is influenced by differences among evaluators, labeling bias, data preparation and prompt formulation, while the limited range of tasks, prompting patterns and datasets, which may not be fully representative or may have been included in the data used to train GPT models, makes it difficult to generalize the results (Ronanki et al., 2024).

Several research gaps have also been identified in the literature. Certain activities, such as terminology extraction, relationship extraction and change impact analysis, remain insufficiently investigated. Regarding input data, structured documents, particularly SRS documents, predominate, while informal and dynamic sources are considerably less represented. A similar pattern can be observed in output results, which are most often focused on traditional Requirements Engineering artifacts, whereas traceability, test artifacts and other more complex outputs are less frequently addressed. This indicates a need for approaches that integrate different types of input and output data and support more comprehensive Requirements Engineering workflows. In addition, by generating explanations of their results, LLMs may facilitate the understanding of decisions in analytical tasks, including requirements classification, defect detection and other analysis activities. (Zadenoori et al., 2025)

Future research may focus on integrating LLMs with other artificial intelligence techniques, formal methods and domain-specific tools, as well as on combining Requirements Engineering knowledge with knowledge from other fields to improve requirements quality. Particular attention should be devoted to less explored activities, such as terminology extraction, relationship extraction and change impact analysis. (Zadenoori et al., 2025) Another important direction is the development of new prompting patterns and their combination to obtain more stable and reliable results, as well as the optimization of the trade-off between precision, recall and accuracy. In addition, new evaluation frameworks should be developed and the boundaries of LLM application in different Requirements Engineering activities should be more clearly established. Their application should be accompanied by an appropriate level of human oversight to ensure the reliability, quality and responsible use of generated results. (Ronanki et al., 2024)

Table 1 presents an overview of the selected LLMs and their applications in various Requirements Engineering activities.

Table 1. *Overview of LLMs and Their Applications in Requirements Engineering*

Study	Model	Specific task	Application and key findings
(Yeow et al., 2024)	**GPT-3.5**	Generation of questions for surveys and interviews in requirements elicitation	Used with short and simple prompts. The quality of the questions was assessed based on readability, clarity, relevance and completeness. The average Flesch score was approximately 40. The quality of the output depended on the use of specialized terminology and prompt formulation.
Krishna et al., 2024)	**GPT-4 / ChatGPT**	1. SRS document generation 2. Correction of problematic requirements	Used to develop an SRS for a student club management portal. It generated more concise documents with less repetition, while for performance requirements it generated fewer but more precise requirements. It was more successful in correcting problematic requirements, correcting 10 out of 11 problematic requirements and showed smaller average deviations from human evaluators' assessments. It contributed to reducing the time required to develop the SRS. Weaker results were obtained for external interfaces compared with the human-written SRS.
Krishna et al., 2024)	**CodeLlama-34b**	1. SRS document generation 2. Correction of problematic requirements	Generated shorter but more detailed, complete and consistent documents, with a larger number of requirements in most sections. It provided more detailed security requirements, while in some categories it achieved results comparable to or better than the human-written SRS. Identified problems included generic, irrelevant and misclassified requirements, copying requirements from other sections and combining multiple requirements into a single requirement, which could negatively affect understandability and lead to ambiguity. It corrected 4 out of 11 problematic requirements and contributed to reducing the time required to develop the SRS. Weaker results were obtained for external interfaces compared with the human-written SRS.
(Lubos et al., 2024)	**Llama 2 70B**	1. Defect detection, 2. Justification of assessments 3. Requirements reformulation	Requirement quality was assessed according to the criteria of the ISO/IEC/IEEE 29148 standard. The model achieved high recall but lower precision due to the identification of non-existent problems. The explanations were generally assessed as convincing, while the proposed reformulations were evaluated as improvements.

(Ronanki et al., 2024)	**GPT-3.5-turbo**	1. Classification of functional and non-functional requirements 2. Identification of relationships between requirements	Performance depended on prompt formulation and temperature. Precision, recall, F1-score and accuracy were used for evaluation. The best and most stable results for requirements traceability were achieved using the "Question Refinement" pattern, while "Cognitive Verifier" and "Question Refinement" were the most successful for requirements classification. The choice of prompt and temperature significantly affected model performance.

CONCLUSION

The aim of this paper was to analyze the application of large language models in requirements engineering, with a focus on the specific tasks in which they are applied, the models used, the methods for evaluating the results and the limitations of their use. The paper first provides an overview of the basic concepts of requirements engineering and large language models, followed by an analysis of their applications in various requirements engineering activities. The applications of LLMs in requirements elicitation, specification, analysis and validation were analyzed, as well as the applications of GPT-3.5, GPT-4, ChatGPT, CodeLlama-34b and Llama 2 70B.

The analysis showed that LLMs are applied across different phases of requirements engineering, from elicitation and specification to analysis and validation. Examples of their application in question generation, specification development, requirements classification, identification of relationships between requirements, defect detection and requirements improvement were considered. It was observed that the choice of model, prompt formulation and evaluation method can significantly affect the obtained results. LLMs can contribute to the automation of certain activities and reduce the time required for requirements processing; however, their application is limited by the possibility of generating inaccurate and biased content, difficulties in understanding context, ambiguous and conflicting requirements, as well as data protection and security concerns. Therefore, human verification and oversight are necessary.

Future research may focus on improving evaluation methods, more effective context management and the development of new prompt formulation patterns. Particular attention should be paid to the application of LLMs to domain-specific requirements, as well as to less explored activities, such as terminology extraction, relationship extraction and change impact analysis. Integrating LLMs with other artificial intelligence techniques, formal methods and domain-specific tools may further contribute to the reliability and efficiency of their application in requirements engineering. Future development should aim to establish a balance between the potential for automation and the need for human oversight in order to ensure the quality, reliability and responsible use of generated results.

ACKNOWLEDGEMENT

This research was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia, with a scholarship for PhD students in the year 2026.

REFERENCES

Arora, C., Grundy, J., & Abdelrazek, M. (2024). Advancing requirements engineering through generative AI: Assessing the role of LLMs. In A. Nguyen-Duc, P. Abrahamsson, & F. Khomh (Eds.), *Generative AI for effective software development* (pp. 129–148). Springer. https://doi.org/10.1007/978-3-031-55642-5_6

Binkhonain, M., & Alfayez, R. (2025). Are prompts all you need? Evaluating prompt-based large language models (LLMs) for software requirements classification. *Requirements Engineering, 30, 423–443.* https://doi.org/10.1007/s00766-025-00451-8

Hemmat, A., Sharbaf, M., Kolahdouz-Rahimi, S., Lano, K., & Tehrani, S. Y. (2025). Research directions for using LLM in software requirement engineering: A systematic review. *Frontiers in Computer Science, 7,* 1519437. https://doi.org/10.3389/fcomp.2025.1519437

Jin, H., Huang, L., Cai, H., Yan, J., Li, B., & Chen, H. (2024). From LLMs to LLM-based agents for software engineering: A survey of current, challenges and future. arXiv preprint arXiv:2408.02479. https://doi.org/10.48550/arXiv.2408.02479

Krishna, M., Gaur, B., Verma, A., & Jalote, P. (2024). Using LLMs in software requirements specifications: An empirical evaluation. *In 2024 IEEE 32nd International Requirements Engineering Conference (RE)* (pp. 475–483). IEEE. https://doi.org/10.1109/RE59067.2024.00056

Lubos, S., Felfernig, A., Tran, T. N. T., Garber, D., El Mansi, M., Erdeniz, S. P., & Le, V. M. (2024). Leveraging LLMs for the quality assurance of software requirements. *In G. Liebel, I. Hadar, & P. Spoletini (Eds.), Proceedings – 32nd IEEE International Requirements Engineering Conference, RE 2024* (pp. 389–397). IEEE. https://doi.org/10.1109/RE59067.2024.00046

Marques, N., Silva, R. R., & Bernardino, J. (2024). Using ChatGPT in software requirements engineering: A comprehensive review. *Future Internet, 16*(6), 180. https://doi.org/10.3390/fi16060180

Nguyen-Duc, A., Cabrero-Daniel, B., Przybylek, A., Arora, C., Khanna, D., Herda, T., Rafiq, U., Melegati, J., Guerra, E., Kemell, K.-K., Saari, M., Zhang, Z., Le, H., Quan, T., & Abrahamsson, P. (2025). Generative artificial intelligence for software engineering: A research agenda. *Software: Practice and Experience, 55(11),* 1806–1843. https://doi.org/10.1002/spe.70005

Ronanki, K., Cabrero-Daniel, B., Horkoff, J., & Berger, C. (2024). Requirements engineering using generative AI: Prompts and prompting patterns. In A. Nguyen-Duc, P. Abrahamsson, & F. Khomh (Eds.), *Generative AI for effective software development* (pp. 109–127). Springer. https://doi.org/10.1007/978-3-031-55642-5_5

Yeow, J. S., Rana, M. E., & Abdul Majid, N. A. (2024). An automated model of software requirement engineering using GPT-3.5. *In 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)* (pp. 1746–1755). IEEE. https://doi.org/10.1109/ICETSIS61505.2024.10459458

Zadenoori, M. A., Dąbrowski, J., Alhoshan, W., Zhao, L., & Ferrari, A. (2025). Large language models (LLMs) for requirements engineering (RE): A systematic literature review. arXiv preprint arXiv:2509.11446. https://doi.org/10.48550/arXiv.2509.11446