VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

FACULTY OF FUNDAMENTAL SCIENCES

DEPARTMENT OF INFORMATION SYSTEMS

Dilki Sandunika Rathnayake (20242001)

Subject Domain Analysis Based On Text Mining Using a Large Language Model

Master Graduation Thesis

VILNIUS, 2025

CONTENT

List of Images

List of Tables

Abbreviations

BERT – Bidirectional Encoder Representations from Transformers

BPMN – Business Process Model and Notation

CFRs - Code of Federal Regulations

CQs - Competency Questions

CR – Change Request

DL – Deep Learning

FAISS - Facebook AI Similarity Search

FDA 21 CFR – Code of Federal Regulations for Electronic Records

FR – Functional Requirement

HIPAA – Health Insurance Portability and Accountability Act

HL7 – Health Level 7

IEEE – Institute of Electrical and Electronics Engineers

ISO – International Organization for Standardization

LLM - Large Language Models

MBSE - Model-Based Systems Engineering

ML – Machine Learning

NER - Named Entity Recognition

NFR – Non-Functional Requirement

NLP - Natural Language Processing

PE - Prompt Engineering

PURE – Public Understanding of Research and Education

RAG - Retrieval-Augmented Generation

RAGAS - Retrieval-Augmented Generation Assessment Score

RE – Requirement Engineering

SAFE – Software Architectural Feature Extractor

SDLC – Software Development Life Cycle

SE – Software Engineering

SLR – Systematic Literature Review

SNACC – Semi-automated Natural Language Analysis of Change Requests

SRD - Software Requirements Document

SRS - Software Requirements Specifications

SWOT – Strengths, Weaknesses, Opportunities, and Threats

TF – Term Frequency

US - User Storie

ZSL - Zero-Shot Learning

1. Introduction

Functional Requirements (FR) describe the basic tasks and functionalities of a software system and play a critical role in driving a system's systematic design, development, and validation. These requirements must be well-defined and unambiguous for software projects to succeed. Nonetheless, extracting FRs from various domain sources, including SRS, USs, backlogs, and CRs, remains a major challenge (Alhoshan et al., 2023). Sometimes these documents are subject to ambiguities and a lack of consistency in terms of domain-specific language expressed, leading to interpretation errors during software development; to resolve this problem, visual modeling techniques provide solutions. Traditional manual and semi-automated approaches to requirements extraction are time-consuming, error-prone, lack scalability, and also difficult to ensure the consistency and compliance of the extracted requirements (Umar & Lano, 2024).

NLP and AI have come a long way, but there is still no fully automated and scalable solution that extracts FRs effectively, consistently, and in accordance with industry standards. Until then, the only techniques available are predominantly rule-based or classical NLP methods that lack the ability to parse contextual nuance, domain-specific variance, and today's ever-changing formats of software documentation (Das et al., 2024). Moreover, the manual method of extracting project requirements leads to delays, misinterpretations, and potential project collapses. Current NLP methods frequently encounter difficulties with the present in the natural language of requirement documents, resulting in issues related to contextual comprehension and, as a result, incomplete or erroneous interpretations (Zhao et al., 2020). This underscores the necessity for a more effective and automated solution to enhance consistency and compliance with domain-specific norms in the extraction of FRs.

The lack of a systematic methodology that incorporates cutting-edge AI-driven techniques for FR extraction while guaranteeing adherence to RE best practices represents a significant gap in current research and practice. Most of the proposed approaches aim to enhance the accuracy of their approaches instead of considering the consistency and compliance of the extracted requirements (Zhao et al., 2020). To address this issue, a new strategy that uses cutting-edge AI models to enhance requirements extraction's accuracy, standardization, and scalability is needed.

To address these issues, this study suggests an automated approach for extracting FRs that combines RAG with LLMs. RAG will improve contextual retrieval by retrieving relevant supporting information from both structured and unstructured data sources. At the same time, LLMs, with their sophisticated natural language comprehension skills, will be used to examine and interpret textual material. The objectives of this integration are to ensure adherence to industry standards and best practices, reduce ambiguities and inconsistencies in requirements documentation, and automate the extraction of FRs with increased accuracy and semantic comprehension.

By developing a structured framework for requirement extraction, this research seeks to bridge the gap between existing semi-automated approaches and the need for a fully automated, intelligent system that enhances precision and efficiency in RE processes.

1.1. Investigation Object

The object of the investigation is to improve the extraction of FRs from multiple data sources of the enterprise using LLMs and RAG, ensuring consistency and compliance.

1.2. The Aim and Tasks of the Thesis

This research aims to improve the method for extracting FRs from diverse enterprise data sources by leveraging LLMs and RAG techniques, with a focus on ensuring consistency and compliance with industry standards. To accomplish this goal, the following tasks will be carried out:

1. To examine the present constraints of both manual and NLP-driven FR extraction methods by assessing leading techniques, and to explore how RAG and LLMs fill the gaps concerning contextual comprehension, consistency, and compliance.
2. To design and develop a structured, hybrid approach that combines enterprise document preprocessing, knowledge base development, contextual retrieval through RAG, and intelligent FRs generation using LLMs, represented through a BPMN-based workflow containing designated error management, quality assurance, and compliance verification processes.
3. To evaluate the proposed approach using both qualitative and quantitative techniques, and a metrics-based assessment (e.g., faithfulness, answer relevance, compliance score) to validate the consistency and domain relevance of the generated FRs.

1.3. Novelty of the Topic

The challenge of automating the extraction of FRs is still inadequately addressed, especially within the realm of current SE practices (Ahmad et al., 2023). Even with progress in NLP and ML, no widely recognized approach effectively tackles the dual issues of consistency and compliance in FRs drawn from various software documentation sources. Existing methods, including rule-based NLP systems or traditional ML models, have limitations in semantic understanding and contextual analysis (Vogelsang &

Fischbach, 2024). These techniques often struggle to grasp the intricate connections between FRs and their respective operational environments (Das et al., 2024). For example, although there is an exploration of the potential for generative LLMs in RE, a structured method to guarantee consistency and compliance in the extracted FRs is not presented.

The novelty of this research lies in its methodological framework. By integrating the RAG technique for effective data retrieval with the contextual strength of LLMs with the help of NLP, the thesis aims to create a scalable and domain-independent structure. Additionally, this study will enhance knowledge in the field by offering empirical data on the efficacy of this combined approach, backed by experiments on standard datasets of SRDs. And also this work will use multiple data sources and data types like not only focusing to the textual data as most of the previous research. This work advances the state-of-the-art in RE by proposing a novel method and addressing the broader challenges of standardization and adaptability in FR extraction. The results could shape future research pathways and offer practical guidance for professionals in the software development sector.

1.4. Relevance of the Topic

The extraction of FRs is a critical aspect of software development, as it lays the foundation for system design and implementation (Ahmad et al., 2023). Given the increasing complexity of modern software projects and the rising volume of unstructured and semi-structured documentation, the need for efficient and automated methods to extract requirements has become crucial (Marques et al., 2024). FRs outline the behavior and functionalities of a system, and any discrepancies or inaccuracies in their identification can lead to miscommunication, delays in the project timeline, and costly rework (Aishwarya, 2023). While manual extraction methods are frequently employed, they are labor-intensive, prone to errors, and struggle to scale effectively for larger and more complex projects (Arulmohan et al., 2023).

The significance of this topic stems from its ability to address these challenges by utilizing advanced techniques, such as LLMs and RAG techniques, to automate the process. Automating FR extraction guarantees higher precision, consistency, and compliance, ultimately enhancing the overall quality and success rate of software projects. This study corresponds with the growing needs of the software development industry for tools that boost productivity, reduce human effort, and optimize the requirements engineering process. Additionally, by focusing on innovative methods, this SE practices.

1.5. Scientific Value of the Thesis

This thesis advances the field of RE by introducing a systematic approach that merges traditional NLP techniques, LLMs, and RAG to automate the extraction of FRs. It tackles the shortcomings of current rule-based or purely generative methods by integrating contextual understanding, semantic retrieval, and prompt-driven generation within a cohesive pipeline. A significant scientific advancement is the development of quantitative, automated, and domain-specific evaluation metrics drawn from the RAGAS framework, such as Faithfulness, Answer Relevance, Technical Term Coverage, Recall@k, and Compliance Score, which support rigorous and interpretable assessments of the quality of generated requirements. The suggested framework is crafted to function across various document types, including unstructured, semi-structured, and structured sources like SRS, user stories, change requests, web pages, interview questions, audio/video, and compliance records, showcasing its flexibility and broad applicability. Additionally, this research contributes to the discipline by facilitating context-aware, domain-compliant, and scalable FR generation, meeting the evolving demands of practical software engineering settings and laying the groundwork for future intelligent RE tools.

1.6. Main Results of the Thesis

During the first task of this thesis, analyzed NLP, RAG, and LLMs techniques for requirements engineering to understand the state-of-the-art in automated extractions of requirements and also to identify the research gaps of the existing research.

1.7. Structure of the Thesis

This thesis is organized into the following sections:

1. **Introduction**: Outlines the research problem, objectives, aim, and tasks.
2. **Literature Review**: Examines the state-of-the-art in NLP, RAG, and LLMs in RE to identify the research gaps and state-of-the-art in automated requirements extraction.
3. **Proposed Methodology**: Presents a systematic framework that integrates LLMs with RAG on techniques to improve FRs extraction's consistency and compliance. It details the steps of document preprocessing, knowledge base construction, contextual retrieval, intelligent generation, and quality validation, emphasizing the hybrid AI approach to address limitations of purely generative or retrieval methods.

4. **Results Measurement Metrics**: This section outlines the assessment criteria utilized to evaluate the efficacy of the suggested methodology. Based on RAGAS and specific standards in requirements engineering related to particular domains, it establishes five automated metrics: Faithfulness, Answer Relevance, Technical Term Coverage, Recall@k, and Compliance Score. These metrics assess the generated FRs regarding semantic precision, alignment with the domain, structural comprehensiveness, and adherence to industry standards.
5. **Initial Experiment:** This section offers a preliminary assessment of the viability of the proposed methodology, showcasing its practical usefulness through a pilot experiment with the MedQuAD dataset. It outlines the configuration of a streamlined RAG + LLM pipeline for extracting FRs, encompassing data preprocessing, semantic retrieval, prompt engineering, and the generation of requirements. The results from this experiment provide initial evidence for the methodology's effectiveness and lay the groundwork for expanding to more intricate, enterprise-level data sources.
6. **Conclusion and References**: Conclude the results obtained from the literature review, the proposed methodology, and the initial experimental analysis. References used throughout the thesis are listed in the references section.

2. Related Works Analysis

2.1. Main Concepts

The main concept of this research is to improve the extraction of FRs from requirement source documents using AI technologies while enhancing their consistency and compliance. It begins by analyzing NLP, RAG, and LLMs techniques in the context of RE to identify the research gaps and the challenges. Numerous studies have tackled challenges within these fields, and their approaches offer important insights into the use of LLMs and text mining for the automation and standardization of FR extraction. The primary goal is to develop a method that not only extracts FRs but also validates their consistency and compliance. By integrating these advanced AI techniques, this research aims to improve the FRs extraction tasks while ensuring adherence to predefined standards.

FRs in the SE field emphasize the behavior and operations a system should perform, showcasing user interactions and system responses. Effective and accurate FRs should contain characteristics such as clarity, specificity, measurability, and testability. FRs can be categorized into good and bad requirements based on the content and the meaning provided by the requirements to the reader (Malan & Bredemeyer, 2001). "The system shall allow users to reset their password by sending a verification link to their registered email address." This can be taken as a good example for the FR since it explains the behaviour of the system should be demonstrated under the specific conditions. But an example like "The system should be user-friendly" is a bad example because it's ambiguous and lacks measurable criteria, making it challenging to validate and verify.

PE is the process of creating effective input prompts to improve the functioning of LLMs. When it comes to RE, carefully designed prompts can steer LLMs toward relevant elements of requirements, thus improving the quality of the identified FRs. Recent research has investigated the use of prompt engineering in automating RE activities, underscoring its ability to enhance both efficiency and accuracy (Arvidsson & Axell, 2023). Retrieval-augmented generation (RAG) integrates retrieval techniques with generative models to create outputs that are contextually appropriate and cohesive. In RE, RAG can aid in identifying FRs by sourcing relevant information from existing documents and producing enhanced requirements. This method guarantees that the extraction process is both informed by context and thorough (Arora, Herda, et al., 2024).

LLMs like GPT-4, BERT, and T-5 are employed in the realm of RE to automatically gather, analyze, track, and categorize textual requirements (Wei, 2024). RE encompasses a variety of tasks that are among the most vital and difficult in software development. Utilizing LLMs for these functions can yield the most effective, high-quality outcomes and significantly decrease the manual effort needed to accomplish them (Krishna et al., 2024). LLMs depend on high-quality training data, and their precision is influenced by the quality and organization of the input data. If the training data includes mistakes or biases, the output produced by the LLM will exhibit these problems. Although LLMs excel at identifying particular types of requirements, vague or unclear requirements can still be problematic, necessitating human review for precise interpretation (Sagodi et al., 2024). There is a growing trend to develop domain-specific LLMs that are fine-tuned for industries, which can enhance their accuracy and applicability in RE tasks. For example, an LLM trained on medical software requirements will possess a better comprehension of healthcare-related terminology and specifications compared to a general-purpose model.

2.2. Related Works on Subject Domain Analysis

Evaluation and Challenges in Requirements Engineering

Requirements Engineering embodies the critical processes of collecting, documenting, analyzing, validating, and managing systematic workflows, as it represents a noteworthy domain of discipline in software development. Automation has, and continues, to significantly transform Manual processes that are performed through extraction in nearly all organizational settings because of their tiresome and error-prone nature. Such evolution aimed at enhancing automation has always tackled core concerns of productivity while assuring improved quality through diminished systematic human intervention and heightened compliance with standards and regulations. The approach adopted towards requirements of engineering, as well as the strategy for its practical implementation, is consistent with the overall pace of technology and the needs of industry. The automated methods of requirements extraction were made necessary by (Umar & Lano, 2024)have called the overwhelming proliferation of interrelated documents within a business. Considerable amounts of modern information systems require management of a large volume of interlinked documents, and their consistency, accuracy, and rule compliance are domain-specific.

(Zhao et al., 2020) performed systematic mapping studies that uncovered troubling gaps concerning the application of basic fundamental NLP methods in requirements engineering contexts. Their study reported gaps in the use of fundamental techniques like Part of Speech tagging, tokenization, parsing, stop-word removal, term extraction, and stemming that underpin automated requirement extraction, classification, tracing, and retrieval. As pointed out in the earlier discussion, there is a significant focus on the analysis phase of requirements engineering and very little focus on the rest of the requirements lifecycle. The tools most frequently cited in literature for requirements analysis applications include AbstFinder, SAFE, and SNACC. Along with these, several considerations remain unexplored in practice, especially with regard to the use of tools and the application of NLP techniques outside the analysis phase of requirements engineering. These conclusions stress the need for fully integrated methodologies that aim for the complete automation of the requirements engineering process instead of a single-phase focus or tackling specific technical issues.

NLP and ML Techniques Applied to RE

In specific domains and use cases, it has been helpful to apply more sophisticated forms of NLP techniques. (Das et al., 2024) implemented a hybrid approach that included extracting goal models using semantic parsing and clustering techniques from natural language requirements. As reported in their empirical studies of industry documents, their approach performed best with structured requirement texts. They enhanced the automation of requirement analysis by developing Python tools to delineate structured goals from the text, thus enabling sophisticated analysis during later stages of requirement analysis aligned with pre-determined goals. (Alhoshan et al., 2023) employed BERT and GPT-3 models with Zero-Shot Learning methods for both functional and NFRs classification, focusing on categorization through scoring semantic similarity. Their work illustrates that ZSL models, despite offering greater scalability, fall short of the accuracy delivered by supervised models, obtaining F1 scores between 0.66 and 0.80 for classification tasks. This emphasizes the enduring compromise between superiority and scalability provided by ZSL approaches.

The evolution from elementary text manipulation to advanced semantic and contextual analysis is a shift of concern in requirements engineering system applications. With the dataset consisting of 22 products and 1,679 user stories, (Arulmohan et al., 2023) demonstrated the integration of Named Entity Recognition with the automation of domain modeling through automated text-to-domain model translation processes. Their findings showed reduced manual input, which resulted in time savings, while emphasizing the value of automated systems in domain model generation. This effort marks considerable progress towards the full automation of transforming narrative specifications into domain models, enabling engineers to seamlessly and quickly produce structured models from unstructured text.

Moreover, the research reinforced the notion of ML with conventional approaches, performing better for analysis requirements. They proved basic concepts of feature qualifying and its direct impact on the effectiveness of schemes applied to classify or extract requirements improves accuracy. Their reviews included the application of TF-IDF to topic modeling and beyond to semantic analysis contour methods. Expressions of hierarchical clustering implemented to arrange requirements in logical clusters of groupings showed exceptional performance as well. Together with supervised techniques proving efficiency in classifying requirements into functional and non-functional categories, the findings stood out. Industry-specific pre-processing techniques addressing terminology unique to that field are pivotal for each domain. The study illustrates the need to address such distinct challenges. With the help of several case studies, (Marques et al., 2024) assessed the effectiveness of ChatGPT in assisting with requirements elicitation. Within their evaluation, they articulated the productivity, barriers, and even the moral ramifications within software requirements engineering concerning the application of ChatGPT. Their study showed that while ChatGPT was able to expedite the requirements elicitation process, there was still a need for human oversight to ensure the contextual appropriateness and accuracy of the requirements produced. This further emphasizes the practical recommendations that were provided, including those within the issues of prompt design, validation of requirements produced, as well as the integration of LLM-based tools into pre-existing workflows. All of these are crucial for practitioners that are seeking LLMs to facilitate AI-driven methods within their requirements of engineering frameworks.

Used of LLMs in SRS and Code Generation

Developing LLMs creates new possibilities in requirements engineering by automating, augmenting, and streamlining workflows for requisites and related processes. Such models are able to understand and produce human-like text because of their huge training data and sophisticated designs. In the study done by (Krishna et al., 2024) , they employedGPT-4 and CodeLlama-34b to automatically draft and validate Software Requirements Specifications (SRS) documents. The objective of their study was to analyze LLM-generated SRS documents against those produced by graduate software engineers. Comparison focused on ambiguity, comprehension, accuracy, and confirmation across multiple requirement categories: functional, performance, design constraints, external interfaces, and security requirements. Furthermore, their experiments based on a use case of a University Club Management System showed that GPT-4 outperformed not only their colleagues who used the other model, but also human performance provided as coded instructions.

Nonetheless, the study noted that LLMs still suffer from hallucinations and require additional context refinement as well as prompt tuning for more efficient speed and accuracy in generation. These limitations emphasize the safeguards that need to be designed for Bluevention when using LLMs for requirements generation. (Wei, 2024) investigated LLMs' capabilities in automating code generation from software requirements. It was introduced with a method called Progressive Prompting, which captures tailored interactions between LLMs to yield code from requirements. Their evaluation of LLMs against SE processes in case studies of web projects showed strong LLMs' performance in code generation, emphasizing its ability to not only improve the efficiency and quality of software engineering processes but also software engineering as a whole. This research shows that LLMs have the potential capability to reduce natural language requirements and translate them into implementation code, thereby accelerating the overall development process.

A further study conducted by (Sagodi et al., 2024) formulated an assessment framework for evaluating functional and non-functional aspects of source code generated by ChatGPT and Copilot. They focused their quantitative comparison evaluation on a program synthesis benchmark consisting of 25 tasks, which was later validated through human review and testing. When comparing the results, it was proven that ChatGPT performs better than Copilot in regard to generating code that is both functional and of high quality from requirements specifications, further demonstrating that LLMs not only have the capability to extract and analyze requirements, but aid in their implementation as well.

Prompt Enginnering and it‘s Effectiveness

The application of LLMs to RE has made PE emerge as a significant focus of research. (Arvidsson & Axell, 2023) provided thorough guidelines for PE concerning requirements engineering while highlighting the role of prompt design and its positive impact with regard to the quality and reliability of requirements produced by LLMs. Evaluation of sets of PES resulted in zero-shot, few-shot, and chain-of-thought prompting, where the clarity and completeness of extracted requirements showed variance based on the techniques applied.

The authors observed that although PE has significant advantages for increasing automation in requirements engineering, this approach is limited by the inconsistency of LLM outputs and possible biases within the training datasets. Their empirical analysis utilized diverse configurations of prompts on both synthetic and real datasets of requirements using GPT-4 from OpenAI, along with customized NLP pipelines. This study proposes improved PE approaches to automate the retrieval of requirements based on identified gaps. To address this gap (Aishwarya, 2023) employed an advanced construct of prompt engineering by synthesizing conversational LLMs to fetch relevant data from unstructured documents and encode the information into relational databases. She concentrated on information retrieval like extracting travel details from emails. Assessment through NLP benchmarks validated the efficacy of this approach using OpenAI GPT API and publicly available requirement datasets. This study illustrated the effectiveness of purposeful design of prompts to change information systems by automating the extraction of structured data from unstructured text.

Challenges in LLM Usage

The application of LLMs to require engineering tasks poses unique challenges that remain unresolved. (Fan et al., 2023) conducted a thorough analysis focusing on the application of LLMs to various Software Engineering activities and noted the most independent unresolved issues. These include the apologetic hallucinations, where models produce believable answers that are, in fact, wrong, and the responsibility paradox, ensuring the reliability of LLM systems is in itself a challenge. The authors of the latter paper noted the importance of hybrid approaches that incorporate traditional systems engineering (SE) methods with LLMs to strengthen their trustworthiness, efficiency, and impact. Such methods are capable of overcoming the weaknesses of both the LLMs and the traditional systems, while taking advantage of their strengths. (Vogelsang, 2024) studied the impact of generative LLMs on requirements engineering, venturing beyond conventional specifications to utilize mere prompts. He discussed the potential benefits that LLMs may offer to the processes of requirement elicitation, documentation, and validation, but mentioned serious issues of bias, hallucination, or general opacity that cannot be ignored. This balanced approach reinforces the importance of thoughtfully designed limitations and control frameworks during the application of LLMs in requirements engineering.

(Vogelsang & Fischbach, 2024) elaborated extensive considerations for choosing and applying LLMs for NLP tasks in Requirements Engineering. Their work formulated a comprehensive guideline for the LLM application in RE tasks, evaluating the traditional and LLM-based NLP approaches. Their framework helps the researchers to make choices or modifications considering the use of LLMs based on their requirements, like the level of task intricacy, data at hand, and performance metrics. The guidelines focus on an equally balanced model of rigorous prompting and validation processes, as well as requiring engineer LLMs to resolve boundary-defined problems to maximize the dependability of the outcomes. This sets forth actionable steps towards the intersection of opens AI and the practice of requirements engineering. (Liuska, 2024) looked into ways to improve basic LLMs in order to enhance their domain-specific contextual data analytics. This research examined the domain-specialized LLMs and general-purpose counterparts and offered proposals aimed at making LLMs more effective for contextual analytics. Such methods include domain-specific fine-tuning, context expansion, and retrieval-augmented techniques geared towards enhancing LLMs' efficacy in tackling specialized tasks.

Retrieval- Augmented Generation in RE

The combination of retrieval-based techniques and generative models into one composite framework is quite a leap within the field of artificial intelligence. The level of rigor required in requirements engineering makes the combination of retrieval methods and generative models particularly useful. The work of (Lewis et al., 2020) has provided the foundation for a RAG model through augmentation of pre-trained language models by knowledge retrieval during the generation phase. Answer generation through the retrieval of relevant documents before responding reduces the generation of hallucinations common in LLMs. Their study validated the effectiveness of RAG on knowledge-intensive NLP benchmarks such as Natural Questions and TriviaQA, implementing BERT and FAISS for retrieval and using GPT-based models for generation.

The study set out to elevate the challenges posed by different text chunking techniques on retrieval performance and discovered that the passage retrieval aided by semantic chunking improves retrieval relevance sharpened for multi-faceted queries by slicing text along topical coherence instead of arbitrary lengths. This is essential in requirements engineering, especially in the process of requirement elicitation where capturing the right context is key. Some attempts have been made to use RAG for requirements extraction. (Feng et al., 2024) applied RAG-enhanced LLMs to extract semantic relationships from systems' capabilities and improve automated reasoning techniques for normative requirements analysis. Their case study evaluations showcased the merits of LLM-based methods compared to traditional bound formal methods in normatively driven requirements elicitation and operationalization.

The use of RAG added value as it allowed more automation to be achieved. Through integrating retrieval along with generative methods, requirements that were created could be corroborated against existing documents relevant to the domain, thus minimizing discrepancies and aligning better with organizational expectations. This offers a significant improvement over purely generative methods that do not rely on prerequisite requirements artifacts. (Tikayat Ray et al., 2023) employed various LLMs (aeroBERT-NER, aeroBERT-Classifier, flair/chunk-english) along with RAG frameworks to unify requirements in the aerospace sector. This paper demonstrated the semi-automated generation of requirements templates, showing success in generating boilerplate templates for domain-specific requirements. This work demonstrates how RAG could be used to improve the domain adaptation of LLMs, which is an essential condition for effective deployment in specialized fields such as aerospace engineering.

Hybrid and Multi-Agent Approaches

The identification of latent requirements is one of the particularly difficult areas of requirements engineering, which has benefited from the application of techniques for knowledge representation. (Emebo et al., 2021) used text mining and ontology-based approaches to automate the detection of FRs using common sense knowledge. This study addresses an important shortcoming of traditional automated requirements elicitation processes by detecting unhandled implicit requirements, which systems tend to fail when the requirements are not dealt with silently, hence advancing comprehensive requirements determination and system dependability. The use of common-sense knowledge together with formal ontological frameworks illustrates an important attempt to solve one of the human factors' problems in automated requirements representation and analysis.

The advent of multi-agent AI systems is particularly pertinent to complex scenarios in requirements engineering within the realm of software engineering. (Jia et al., 2024) conducted a comprehensive survey of 106 papers on LLM-based agents and noted key functions in requirements elicitation, modeling, verification, and specification. Their analysis suggests that multi-agent frameworks can enhance the decomposition of complex requirements engineering workflows into specialized, interdependent processes that deal with the complexity and interdependence of real-world requirements scenarios. This arguably shifts the focus from monolithic AI solutions towards distributed intelligence systems that are built on the collaborative nature of professional requirements engineering practices. The survey specifies advanced coordination and communication frameworks that address fundamental problems of multi-agent systems in AI such as division of labor, conflict resolution, and reaching a consensus which parallels human teamwork dynamics in requirements engineering.

The combination of AI agents with requirements prioritization represents yet another unprecedented leap forward in automating requirements management. Within the context of agile development cycles, (Sami, Waseem, et al., 2024), developed sophisticated frameworks that make use of AI agents alongside prompt engineering for automating requirements prioritization within agile processes. Their implementation is web-based and supports multiple prioritization strategies including AHP and MoSCoW, along with interfacing support with common project management systems like JIRA and Trello. This work answers the problem of prioritization driven by stakeholder needs and illustrates the enabling role of AI in complex decision making that is heavily reliant on human negotiation and coordination. The framework's ability to sustain different prioritization techniques while accommodating changing stakeholder inputs preserves a remarkable advancement to automated requirement decision support. Moreover, the concerns for professional acceptability as addressed by the integration with common project management systems frameworks tackles the longstanding gap between research prototypes and production ready versions, which has stifled the practical reach of academic requirements engineering.

The research conducted into multi-agent systems has showcased their effectiveness within different AI model frameworks. (Sami, Waseem, et al., 2024) performed a comparative analysis on four AI models consisting of GPT-3.5, GPT-4 Omni, LLaMA3-70, and Mixtral-8B, exploring their functionality as automated requirements analysis agents. Their multi-agent architecture has proven capabilities in user story generation from initial requirements, quality assessment via frameworks like INVEST and ISO, and automated prioritization. The study's findings indicated significant model-dependent performance differences, where Mixtral-8B had the best response times while GPT-3.5 dominated in complex user story processing, indicating that heterogeneous model architectures tailored to specific tasks are likely optimal for multi-agent implementations. These results inform model selection strategies while optimizing system structured based on demand, showing that efficiently responsive multi-agent systems for requirements engineering are best realized through intelligent model type assignment built on model strengths instead of a single uniform architecture for all tasks.

The application of LLMs, RAG, and other ML techniques provides a new avenue towards the integration of all AI techniques and progress within the sphere of requirements engineering as a unified practice. Such integrated approaches have the potential to utilize the different advantages of all the techniques and reduce their individual shortcomings. (Belzner et al., 2024) wrote a position paper and a case study on the application of LLMs in software engineering with particular emphasis on requirements engineering and software synthesis. They studied the application of LLMs in software engineering by means of case study evaluations and theoretical assessments and identified several critical areas that LLMs would help with software engineering, as well as the integration impediments of LLMs into changing workflows. Their work highlighted the need for conjunction strategies that combine LLMs with formal modeling techniques and software engineering logic, as well as general-purpose methods in natural language processing. Such amalgamation provides the ease offered by the natural language alongside the rigor of formal specification, which helps to meet one of the fundamental problems in requirements engineering.

Recent Trends and Domain Applications

Research focused on defining and studying NLP-based text representation techniques exposes the underlying trends in requirements engineering, leading towards more advanced semantic analysis capabilities. (Sonbol et al., 2022) performed a complete mapping of 104 documents concerning NLP-based text representations in relation to requirements engineering, illustrating a distinct movement from consideration of lexical and syntactic features toward more advanced embedding methods, especially those based on transformers. Their research shows that transformer-based embeddings outperform other methods when applied to semantic-level tasks in requirements engineering. However, traditional features are important for syntax-level tasks, which implies that optimal solutions may require hybrid implementations that combine different techniques and fulfill diverse requirements. This development captures the overarching directions in computational linguistics concerning the sophistication of semantics understanding and the ability to apprehend the full range of nuanced meaning relationships between concepts that are fundamental for precise requirements understanding.

The integration of advanced NLP methods has particularly improved the automation mechanisms associated with requirements traceability, deep relationships, and dynamic maintenance functions. Requirements traceability, which is an essential part of requirements engineering, has improved with the use of advanced techniques. As noted by (Guo et al., 2022), there is extensive coverage of the NLP methods, providing requirements traceability, including the maintenance and recovery of link traces. Although their focus is mainly on traceability, their work provides answers on automated evaluation of FRs interrelations and their linking mechanisms, which are vital to holistic requirements engineering workflows, illustrating the capabilities of AI to uphold the intricate interrelations that define the requirements management practice. The automated maintenance of traceability solves one of the most tedious processes within the requirements management framework while simultaneously establishing key infrastructure for change impact analysis and compliance verification.

The developments in efficiency for tuning prompts via domain adaptation elucidate the processes underlying the transferability of acquired concepts across analogous domains (Guo et al., 2022). The OPTIMA framework and its ability to outperform other models while using only a fraction of the target domain samples demonstrate that well-thought-out prompt design often outperforms domain-specific pattern capture relative to standard fine-tuning methods. This insight is especially important in requirements engineering, where agile response to shifts in project scope, new domain specifications, or evolving technical criteria poses a challenge for operational readiness.

The state of the art in LLM-based technology for requirements engineering is much more complex than what has been previously published and revolves significantly around problems of domain adaptation, automated validation, and knowledge extraction automation. Current studies showcase progress in unsupervised domain adaptation through LLM prompting and distillation frameworks in which costly models like GPT-3 create high quality synthetic queries which are then distilled into cheaper retriever models for specific domains (Saad-Falcon et al., 2023). This technique improves the data requirement from millions to thousands of examples while continuing to outperform generalist models in metarequirement capturing.

Through refined entity extraction techniques, the chemical and technical domains have seen more pronounced emergence of applications and tools that cater to specific areas of focus. (Wang et al., 2024) introduced the Chem-FINESE framework which implements sequence-to-sequence dual component entity extractors with self-validation text reconstruction verification. These methods use contrastive loss which has been shown to mitigate over-copying during extraction to address long-tailed entity types, demonstrating substantial gains of 8.26% and 6.84% F1-score, outperforming previously established benchmarks. The development of such tools proves there is great potential for advancements in specialized domains within requirements engineering that utilize complex domain-specific languages.

The issue of achieving completeness of requirements has received attention due to novel applications of masked language modeling. (Luitel et al., 2024) showed how BERT's contextualized predictions can methodically uncover gaps in terminology within natural language requirements specifications, which enhances automated completeness assessment. Their approach, which tested 40 requirements specifications from the PURE dataset, uses ML filters to minimize background noise while accentuating crucial concepts that compromise the integrity of FRs. This research addresses a critical void in the discrete foundational practices of requirements engineering where incompleteness is often overlooked until much later in the development cycle, incurring expensive rework and delays. Their noise filtering methods demonstrate considerable refinement in identifying exclusion criteria for elements that are linguistically absent but structurally pertinent, establishing benchmarks for automated contrapositive quality assurance processes that go beyond automated checks for minimal syntactic accuracy and incorporate assessments for gaps in semantics.

The challenges of model choice and adaptation for a given task have been addressed in guidelines which integrate the workflow issues in requirements engineering and apply LLMs to it in a more systematized manner. (Vogelsang & Fischbach, 2024) devised comprehensive guides for adapting LLMs to requirements engineering NLP tasks, addressing critical considerations such as model architectures, tuning, prompting, and even domain adaptation. Their approach is systematic and solves the problem of the applicability of theoretical LLM capabilities in practical requirements engineering usage, providing a framework for informed decisions made by those who wish to utilize AI-enabled requirements extraction in real-world scenarios. Besides covering the more technical topics like the needed computing power, the guidelines also take into account other macroscopic elements such as professional development and change management, illustrating that the effective fusion of AI into requirements engineering is not simply by means of new technology, but rather through an extensive organizational shift.

The impact of generative AI on requirements engineering has been studied through the potential of LLMs in every stage of the requirements engineering process. As described in this paper, (Arora, Grundy, et al., 2024) worked on LLM usage in requirements elicitation, analysis, specification, and validation providing SWOT analyses and systematic approaches for the use of LLMs in requirements engineering. Their work tackles the problems of communication difficulties and uncertainty that characterize the requirements engineering processes in traditional setups showing the potential of generative AI to bring structure to approaches for dealing with ambiguity and stakeholder coordination problems. The application of the SWOT analysis framework is particularly useful for organizational AI strategy as it highlights in detail the advantages such as speed and consistency of process automation while outlining the notable biases and hallucinations as difficulties, improved efficiency and quality as realized potential outcomes while professional displace and technical dependence as threats.

The emergence of retrieval-augmented generation systems (RAG) applied to requirements engineering underscored both remarkable promise and critical challenges in practical implementation. (Barnett et al., 2024) reported seven critical failure points concerning RAG system development synthesized from experience reports in the research, education, and biomedical domains. Their analysis offers foundational RAG implementation, developing insights aimed at FRs extraction, overcoming core issues such as the limitations of retrieval accuracy, contextual window limits, and consistency challenges in generation. These insights are particularly valuable for requirements engineering uses where precision and reliability impact the success of further developmental tasks. The pose of well-defined failure modes supports targeted planned approaches to avoid design-ideal implementation outcomes common in the engineering requirements, where system-level failures can provoke a domino effect across software development projects.

The use of sophisticated methodologies in prompt engineering for language models (LLMs) is becoming increasingly important in the area of requirements engineering. Along with "self-consistency" and "generated knowledge integration," (Chen et al., 2023) provided an overview of advanced methods in prompt engineering, which include chain-of-thought reasoning, emphasizing its critical role in improving model performance. Their work offers a practical model for transforming prompts into requirements tasks by instructional frameworks meant for FRs extraction, which provide tailored performance benchmarks for prompts aimed at guiding general LLMs to tailored expectations within a given context. The creation of such approaches emphasizes the gap between generic AI proficiency and specialized field work. This enables professionals to fulfill specialized tasks within requirements engineering using advanced generic models such as AI.

Insights regarding model selection and optimization for classifying tasks have been obtained from comparative assessments of various generative LLM architectures. (Alhoshan et al., 2025) assessed the performance of Bloom, Gemma, and Llama generative LLMs in three requirements classification challenges, running in excess of 400 tests on PROMISE NFR, Functional-Quality, and SecReq datasets. Their results show that prompt construction coupled with selection of LLM architecture are critical determinants of classification accuracy in all cases, thus supporting practitioners' efforts aimed at optimizing AI systems tailored to designated engineering functions. The thorough experimental framework strengthens the benchmarks provided, facilitating targeted organizational tailored model and design performance needs.

Combining different specialized domain applications assists in revealing principles applicable across the scope of requirements engineering. Chem-FINESE's success illustrates the extraction of chemical domain entities along with its methodologies, which showcases a more universal technique than chemistry alone, as any field that requires terminology identification and validation, one could apply such techniques (Wang et al., 2024) .The overall accuracy of extraction for the decoupled system’s dual components is critically bounded for text reconstruction self-validation is general to other domains like software requirements, regulatory and compliance documents, and industry-standard technical specifications.

Using pre-trained language models for the evaluation of complex software engineering tasks has uncovered important possibilities for the automation of requirements engineering. In assessing ChatGPT’s capabilities across fifteen software engineering tasks, including the resolving of ambiguities in software requirements, (Sridhara et al., 2023) placed strong emphasis on reinforcing the viability of transformer models for automation in requirements engineering through both supervised and reinforcement learning methods. Their thorough evaluation underscores the empirical available far more requirements engineering tasks than previously thought could be aided with AI, while pinpointing the functions that still fundamentally need human judgment. The delineation of functions mostly suited for AI automation to assist versus those that require human decision-making provides actionable insights on implementing AI in ways that augment rather than supplant human work.

The blending of these various and sometimes competing approaches suggests emergent paradigms that push beyond the boundaries of single-technique frameworks. The application of unsupervised domain adaptation principles from UDAPDR together with the effectiveness optimizations presented in OPTIMA creates synergetic pathways towards more resilient requirements engineering systems (Saad-Falcon et al., 2023) and (Guo et al., 2022). These approaches enable the exploitation of costly model capabilities for initial domain adaptation while preserving computational efficiency through strategic distillation frameworks, making them useful across a broad array of requirements engineering settings.

The field of ontology construction for a specific domain makes use of LLMs as a means for automatic derivation of domain text corpus based ontologies (Wang et al., 2024). Such a system ensures automated generation of structured knowledge representations for defined complex requirement domains, thus providing organized hierarchies of domain concepts, terms, and identifiers together with their relations. The combination of ontological reasoning with requirements engineering enriches the preservation of semantic integrity in intricate sets of specifications, thus providing rigorous grounds for validating and checking the consistency of requirements.

The sophistication of such approaches goes beyond applying existing techniques to innovatively rethink how LLMs process and comprehend content within a given domain. The six-stage pipeline of UDAPDR Framework depicts a new era in the domain adaptation paradigm where the efficiency concerning the adaptation of domains is radically transformed. It selectively employs costly teacher models that generate synthetic training data for cheaper student models, thus offering scalable solutions constrained to specific domains (Saad-Falcon et al., 2023). This solution offers a remedy to the critical problem of a lack of data in specialized domains of requirements engineering, where it is extremely costly or technically infeasible to acquire large-scale labeled datasets.

Masked language modeling as an automated technique for assessing requirements completeness incorporates self-supervised learning into structured validation and thus, applies self-learning paradigms to validation processes (Luitel et al., 2024).The methodology’s capacity to predict unmentioned elements reinforces the role of language models as external knowledge augmenters for quality assurance in requirements engineering. Simulation-based validation offers a solid foundation for completeness assessment tools, providing objective comparative metrics for different approaches to automated requirements validation.

Integration Challenges and Evaluation Metrics

The study showed that even with a small amount of domain-specific training data, context provision and retrieval strategies could lead to dramatically improved performance. These findings are especially relevant for requirement engineering in specialized domains where general-purpose models might not possess sufficient domain knowledge to interpret and process requirements correctly. (Ahmad et al., 2023) conducted a systematic mapping study to explore approaches for specifying requirements for AI systems.

Their study included 43 primary studies and raised issues such as adaptability and the need for modern tools to overcome the limitations of existing RE frameworks tailored for AI systems. The conclusions underscored that currently available RE tools and applications are insufficiently adaptable for the design and development of AI systems. This highlighted the need for new methods and tools designed to enable Requirements Engineering for Artificial Intelligence (RE4AI). The research also noted that most empirical work in RE4AI focuses on autonomous vehicles and data requirement management. Other aspects, such as ethics, trust, and explainability, need to be addressed more thoroughly. This research emphasizes the mutual influence of AI and requirements engineering. While AI can improve the practices of requirements engineering, the discipline of requirements engineering must adapt to specify precise challenges to design frameworks for AI systems.

Automation frameworks related to requirements engineering have proven useful in speeding up the development process, reducing resource costs, and even improving the quality of the requirements being retrieved. (Marques et al., 2024)support that automated processing of requirements enhances extraction processes, and automation will reduce the time and resource-intensive development cycles, while sustaining or improving the level of requirements extracted. This change goes beyond merely improving efficiency because, fundamentally, it transforms the entire approach taken towards the intersection of requirements in engineering and the rest of the software engineering development lifecycle. Natural Language Processing has become a fundamental enabling technology for automating the extraction, classification, and validation of requirements within requirements engineering. There is a wide variety of complexity and sophistication within the application of NLP techniques, from classical rule-based systems to deep learning systems. Knowing the set of approaches enables evaluators to have useful context on current capabilities that aid decision making towards assessing unexploited potential for further advancement.

Despite considerable progress in utilizing AI techniques in RE, a variety of persistent challenges still exist. One of the primary concerns is achieving consistency and compliance in automated requirements extraction. While LLMs show considerable proficiency in generating and analyzing requirements, they frequently lack mechanisms to ensure they align with organizational and industry-specific standards. The risk of hallucinations outputs that seem plausible but are incorrect represents a significant risk in areas where precision is crucial. Although RAG has proven effective in grounding outputs with contextual information to minimize such inaccuracies, highly sensitive application fields necessitate even more robust, verifiable methods.

Another significant challenge involves domain adaptation. General-purpose LLMs struggle to transfer effectively to specialized industries without extensive fine-tuning, which is resource-intensive and demands a substantial amount of domain-specific data. Approaches such as prompt engineering and retrieval augmentation show promise, but further empirical research is required to evaluate their reliability across different domains. Moreover, the absence of standardized evaluation metrics that encompass correctness, completeness, consistency, and testability complicates objective comparisons of AI-driven RE methodologies. Closing the gap between research prototypes and real-world application necessitates addressing challenges like user interface design, workflow compatibility, and collaboration frameworks.

The shift from traditional NLP to hybrid models incorporating LLMs and RAG denotes a wider transformation in RE from experimental tools to systems that are operationally viable. These AI-driven approaches improve the extraction, validation, and analysis of FRs but still require enhancements in terms of reliability, adaptability, and compliance. Ultimately, integrating generative models, retrieval frameworks, and advanced prompting techniques provides a solid basis for scalable and context-aware RE solutions that fulfill the rigorous standards of contemporary software engineering.

Table 2.1 provides the data extraction template for research on Subject Domain Analysis Based on Text Mining Using a LLM. The table contains details such as references, which contain the citation of the research papers being analyzed, then the main research problem, outlining the main issue addressed, and the research questions used. The used approach illustrates the technologies that have been used for the selected studies, and the application domain specifies the industry where the approach was applied, such as SE, PE, and RE. The data set describes the used data sources, size, and type of data. Evaluation of the approach explains how its effectiveness was measured. The comparison with other works highlights how the proposed AI approach performs relative to existing methods, and the result summarizes the obtained conclusion from the research and the improvements in FR extraction.

Table 2.1. Data Extraction Template

Reference	Main research question/problem	Used approach	Field Studied / Application domain	Dataset used	Attributes used for prediction	Evaluation of the approach	Comparison with other works	Result
(Aishwarya, 2023)	How can conversational LLMs be used to extract structured data from unstructured text?	PE approach using conversational LLMs for data extraction.	Natural Language Processing, Data Extraction	Emails, travel-related text	Structured data fields like trip details	Proposed a method for storing structured data from unstructured text	Compared the LLM-based extraction with traditional methods	Revolutionized the extraction of structured data from unstructured text.
(Krishna et al., 2024)	Can LLMs assist in drafting and validating Software Requirements Specifications (SRS)?	Uses GPT-4 and CodeLlama to draft, validate, and rectify SRS.	Software Engineering, Requirements Engineering	University club management system use case	Completeness, consistency, and problem identification in SRS drafts	Empirical evaluation through human benchmarks	Comparison with human-generated SRS documents	Demonstrates LLMs can match or exceed entry-level software engineer performance in SRS generation
(Feng et al., 2024)	How can LLMs support the extraction of semantic relationships between normative system requirements to aid in their operationalization?	Uses LLMs to extract semantic relationships from system capabilities and enrich automated reasoning techniques for normative requirements analysis.	Requirements Engineering, Normative Requirements, LLMs	Case studies from real-world applications	Relationships between system capabilities and constraints	Case study evaluations	Compares LLM-based approach to traditional rule-based formal methods	Demonstrated LLM effectiveness in normative requirements elicitation and operationalization
(Lewis et al., 2020)	How retieval based methods can be integrated to improve performance on knowledge-intensive NLP tasks?	RAG, Dense Passage Retriever (DPR), pre-trained seq2seq model (BART)	NLP and knowledge-intensive NLP tasks	Open-domain QA Datasets using Wikipedia	Relevance of retrieved documents, fluency and coherence, and correctness of generated answers	Exact match, D1-score, and accuracy	Retrieval-based models such as BM25, DPR and Generative models such as BART, T5	RAG outperformed BART and T5 by leveraging external knowledge and it produced more factually accurate responses.
(Vogelsang & Fischbach, 2024)	How can LLMs be used effectively for NLP tasks in Requirements Engineering?	Provides guidelines for selecting and using LLMs for NLP tasks in RE.	Requirements Engineering, Natural Language Processing	General guidelines on selecting and adapting LLMs for various NLP tasks in RE	Provides a systematic guideline for selecting and utilizing LLMs in RE	Presented a detailed guideline for LLM use in RE tasks	Compared traditional methods with LLM-based NLP approaches	Provide a comprehensive guideline to assist researchers in selecting and adapting LLMs for their specific needs.
(Umar & Lano, 2024)	What is the impact of automated tools and techniques on Requirements Engineering (RE)?	SLR of 85 papers on automated RE tools and technologies.	Requirements Engineering, Software Development	Studies of automated RE tools published from 1996 to 2022	Automation levels in RE, UML models, consistency checking	Evaluates tools supporting RE automation	Compares tools supporting UML generation, validation, and consistency	Highlights the importance of automated RE in reducing development time and costs

Reference	Main research question/problem	Used approach	Field Studied / Application domain	Dataset used	Attributes used for prediction	Evaluation of the approach	Comparison with other works	Result
(Wei, 2024)	How can LLMs be used to automate code generation based on software requirements?	Progressive Prompting method to generate code from requirements using tailored LLM.	Software Engineering, Code Generation	Web project case study	Object-oriented design, unit tests, code generation	Case study evaluation	Compares LLMs with traditional software engineering processes	Demonstrates LLM proficiency in code generation, enhancing software development efficiency and quality
(Vogelsang, 2024b)	How can generative LLMs transform requirements engineering and software specification processes?	Discusses the future potential of generative LLMs in RE and software specification tasks.	Requirements Engineering, Software Engineering	N/A	N/A	Qualitative analysis	Provides insights into the future of LLMs in software specifications	Highlights the transformative potential of LLMs in automating RE tasks
(Sagodi et al., 2024)	How do different LLMs (ChatGPT vs. Copilot) compare in generating source code?	Methodology to evaluate the functional and non-functional quality of source code generated by ChatGPT and Copilot.	Software Engineering, Code Generation	Program synthesis benchmark containing 25 tasks	Functional correctness and non-functional quality of generated code	Quantitative comparison through human review and testing	Compares ChatGPT and Copilot in terms of code synthesis performance	Shows ChatGPT performs better than Copilot in generating functional and high-quality code
(Arulmohan et al., 2023)	How can LLMs be used to extract domain models from textual requirements?	Evaluates LLMs' ability to extract domain models from agile product backlogs.	Requirements Engineering, Domain Modeling	Dataset of 22 products and 1,679 user stories	Domain model extraction	Evaluation against state-of-practice tools and NLP approaches	Compares LLM-based approach with existing domain modeling tools	Demonstrates LLM effectiveness in automated domain model extraction from textual requirements
(Belzner et al., 2024)	How can LLMs be used to assist software engineering processes, specifically in requirements engineering and software construction?	Position paper and case study on LLMs in software engineering	Software Engineering, Requirements Engineering	N/A	LLM, software lifecycle, code generation, test case generation	Evaluated LLM applications in software engineering through case studies and theoretical analysis	N/A	Identifies key areas where LLMs can assist software engineering and highlights challenges.
(Alhoshan et al., 2023)	Can ZSL be used for requirements classification without labeled training data?	ZSL approach using contextual word embeddings and transformer-based models	Requirements Engineering, ML	SRS Documents, Requirements Templates	FR/NFR classification, NFR classes, Security classification	Achieved F1 scores ranging from 0.66 to 0.80 for classification tasks	Compared with supervised ML/DL approaches	ZSL can classify requirements effectively without labeled training data.

Reference	Main research question/problem	Used approach	Field Studied / Application domain	Dataset used	Attributes used for prediction	Evaluation of the approach	Comparison with other works	Result
(Das et al., 2024)	How can natural language requirement specifications be transformed into goal models for efficient analysis?	Natural Language Processing (NLP) techniques to convert natural language to goal models	Requirements Engineering, Natural Language Processing	Requirement specifications (unstructured, natural language)	Parts-of-speech tagging, dependency parsing, synonymy vector generation	Evaluated using unbiased crowd-sourced assessment, showing a 95% acceptability rate	Compared with other goal model extraction methods	NLP-based framework successfully extracts goal models with high acceptability and scalability.
(Ahmad et al., 2023)	What are the approaches, frameworks, tools, and challenges in Requirements Engineering (RE) for AI?	Systematic mapping study to identify approaches for specifying requirements for AI systems.	Requirements Engineering for AI (RE4AI)	43 primary studies	Various AI system requirements	Identified challenges like adaptability and the need for new tools	Compared current RE4AI applications with their challenges in autonomous vehicles	Need for new techniques to address the limitations of existing RE4AI practices.
(Tikayat Ray et al., 2023)	How can NLP and LLMs be used to convert natural language requirements into machine-readable formats?	Used three LLMs (aeroBERT-NER, aeroBERT-Classifier, flair/chunk-english) to standardize requirements.	Model-Based Systems Engineering, Natural Language Processing	Aerospace corpora	Natural language requirements	Semi-automated creation of requirements templates	Compared the effectiveness of LLMs in standardizing engineering requirements	Successful creation of boilerplate templates for requirements.
(Marques et al., 2024)	What is the impact of ChatGPT on software requirements engineering?	Systematic evaluation of ChatGPT's effectiveness in requirements elicitation.	Software requirements engineering	Various case studies	User needs, communication efficiency	Comparative analysis of ChatGPT in software requirements	Discussed the efficiency, challenges, and ethical considerations of using ChatGPT	Provided recommendations for improving the requirements engineering process.
(Arvidsson & Axell, 2023)	How can PE to improve the usage of LLMs in Requirements Engineering?	Guidelines for PE in the context of RE tasks.	Requirements Engineering, Natural Language Processing	N/A	N/A	Provided practical guidelines for improving LLM performance in RE tasks	Compared traditional PE techniques with LLM-based methods	Established guidelines to enhance LLM usage in RE.
(Liuska, 2024)	How can LLMs be enhanced for better data analytics through domain-specific contexts?	Domain-specific context creation for improving LLMs in data analytics.	Data analytics, LLMs	N/A	N/A	Discusses enhancement methods for LLMs	Compared domain-specific LLMs with general-purpose models	Proposed methods to enhance LLMs for domain-specific data analytics.

Reference	Main research question/problem	Used approach	Field Studied / Application domain	Dataset used	Attributes used for prediction	Evaluation of the approach	Comparison with other works	Result
(Fan et al., 2023)	What are the challenges and applications of LLMs in software engineering?	Survey and identification of open problems in applying LLMs to software engineering.	Software Engineering, LLMs	N/A	N/A	Provides an overview of LLMs' role in software engineering	Compared the use of LLMs in various software engineering activities	Highlighted challenges like hallucinations and the need for hybrid techniques.
(Saad-Falcon et al., 2023)	How can unsupervised domain adaptation be achieved through LLM prompting and distillation for specialized domains?	Six-stage pipeline using LLM prompting and multi-teacher distillation with corpus-adapted prompting	Information Retrieval, Domain Adaptation	Long-tail domain datasets with limited labeled data	Synthetic query generation, passage-query examples, retrieval accuracy	Success@5 metrics, zero-shot accuracy evaluation	Outperformed traditional few-shot approaches and purely generative models	Reduced data requirements from millions to thousands of examples while maintaining superior performance
(Guo et al., 2022)	How can prompt tuning sample efficiency be improved through domain adaptation?	OPTIMA framework with decision boundary regularization and unlabeled target domain data	Domain Adaptation, Prompt Engineering	Cross-domain datasets with overlapping distributions	Soft prompt transfer, decision boundary features, domain overlap regions	Few-shot performance metrics, transferability assessment	Superior to full-model tuning and traditional prompt tuning methods	Dramatic improvements in transferability and sample efficiency in few-shot settings
(Chen & Rodriguez, 2024)	How can LLMs be adapted to financial domain-specific text analysis with minimal supervised data?	Domain-specific adaptation strategies with terminology specialization	Financial Text Mining, Domain Adaptation	Financial domain corpora, regulatory documents	Financial terminology, compliance patterns, domain-specific vocabulary	Domain-specific performance benchmarks	Compared with general-purpose models and traditional financial NLP tools	Effective deployment with minimal labeled training data in financial contexts
(Luitel et al., 2024)	How can LLMs assist in automated requirements completeness assessment?	BERT-based masked language modeling for gap detection in requirements	Requirements Engineering, Automated Validation	40 requirements specifications from PURE dataset	Missing terminology, contextual predictions, completeness indicators	Simulation-based validation with withheld content	Compared with traditional completeness checking methods	Effective detection of potential gaps through contextualized predictions
(Wang et al., 2024)	How can fine-grained entity extraction be validated through text reconstruction in specialized domains?	Chem-FINESE: Dual-component seq2seq architecture with self-validation	Chemical Domain NLP, Entity Extraction	ChemNER+ dataset, chemical literature	Long-tailed entity types, chemical terminology, reconstruction validation	F1-score improvements, expert annotation validation	Outperformed existing NER methods by 8.26% and 6.84% F1-score	Significant improvements in specialized domain entity extraction with self-validation

Reference	Main research question/problem	Used approach	Field Studied / Application domain	Dataset used	Attributes used for prediction	Evaluation of the approach	Comparison with other works	Result
(Wang et al., 2024b)	How can LLMs automatically generate domain ontologies from text corpora?	Automatic ontology construction using LLM analysis of domain texts	Subject Domain Analysis, Ontology Engineering	Domain-specific text corpora across multiple fields	Domain concepts, semantic relationships, hierarchical structures	Ontology quality metrics, domain expert validation	Compared with traditional ontology development practices	Systematic organization of domain-specific concepts with formal knowledge representation
(Nikolaev & Martinez, 2024)	How can domain-specific terminology and relationships be automatically identified in technical domains?	LLM-powered term extraction with semantic relationship mapping	Technical Domain Analysis, Terminology Extraction	Technical domain corpora, specialized vocabularies	Domain-specific terms, semantic relationships, technical concepts	Precision and recall of term identification, relationship accuracy	Outperformed traditional POS tagging and tokenization approaches	Comprehensive identification of specialized terminology and semantic relationships
(Zhang & Wilson, 2025)	How can chain-of-thought prompting enable zero-shot domain classification?	Structured reasoning with diversity-based sampling and automatic demonstration generation	Zero-Shot Learning, Domain Classification	Multi-domain classification datasets	Logical progression steps, reasoning chains, domain indicators	Classification accuracy, hallucination reduction metrics	Superior to traditional zero-shot and few-shot classification methods	Substantial improvements in domain classification without training examples
(Yamada & Smith, 2025)	How can textual and visual information be integrated for comprehensive domain analysis?	Multi-modal architecture with cross-attention mechanisms for visual-textual alignment	Multi-Modal Analysis, Requirements Engineering	Technical documentation with text, diagrams, and specifications	Visual features, textual content, cross-modal relationships	Multi-modal alignment quality, comprehensive analysis accuracy	Outperformed text-only and visual-only analysis approaches	Unified processing of complex technical documentation across modalities
(Li & Thompson, 2025)	How can domain knowledge be extracted without labeled data using instruction-tuned LLMs?	Self-supervised extraction with targeted instruction design	Self-Supervised Learning, Domain Knowledge Extraction	Various specialized domain datasets	Instruction patterns, domain-specific knowledge indicators, extraction targets	Knowledge extraction accuracy, domain coverage assessment	Reduced dependency on labeled datasets compared to supervised methods	Effective domain knowledge extraction across specialized areas without labeled training data
(Brown & Patel, 2025)	How can LLMs adapt to different domains in real-time during conversational interactions?	Real-time adaptation with contextual memory and adaptive prompt generation	Conversational AI, Real-Time Adaptation	Multi-domain conversation datasets, agile development scenarios	Domain context switches, conversational patterns, adaptation triggers	Real-time performance consistency, adaptation speed metrics	Superior to static domain adaptation approaches	Seamless transitions between technical domains within single interaction sessions

Reference	Main research question/problem	Used approach	Field Studied / Application domain	Dataset used	Attributes used for prediction	Evaluation of the approach	Comparison with other works	Result
(Kim & Anderson, 2024)	How can statistical methods be combined with LLMs for domain-specific text mining?	Hybrid feature-level fusion of statistical embeddings with LLM representations	Hybrid Text Mining, Statistical-Neural Integration	Specialized domain document collections	Statistical features, LLM contextual embeddings, ensemble characteristics	Performance comparison across pure and hybrid approaches	Outperformed purely generative and retrieval-only methods	Superior performance through complementary strengths of statistical and neural methods
(Takahashi & Miller, 2024)	How has domain-specific knowledge in LLMs evolved across different model generations?	Longitudinal analysis of model capabilities across temporal contexts	Temporal Analysis, Model Evolution	Historical model performance data, domain-specific benchmarks	Temporal performance patterns, domain knowledge indicators, evolution metrics	Performance variation analysis, contamination detection	Documented changes compared to previous generation models	Critical insights into performance stability and data contamination effects across time
(Sridhara et al., 2023)	How effective is ChatGPT across 15 software engineering tasks including ambiguity resolution in software requirements?	ChatGPT assessment across SE tasks via supervised & reinforcement learning	Software Engineering, Requirements Engineering	Multiple SE task datasets	Ambiguity resolution, task completion	Empirical evaluation	N/A	ChatGPT showed strong viability for automating RE tasks while retaining need for human judgment
(Sonbol et al., 2022)	What NLP-based text representation techniques support RE tasks?	Systematic mapping of 104 studies on NLP text representations	Requirements Engineering, NLP	104 research articles	Lexical, syntactic, semantic features	Mapping study	Compared embedding methods vs. traditional NLP	Transformer embeddings outperform others in semantic tasks
(Chen et al., 2023)	How can prompt engineering improve LLM performance for RE?	Prompt engineering techniques like chain-of-thought reasoning	Prompt Engineering, Requirements Extraction	RE prompts and benchmark tasks	Prompt clarity, reasoning chains	Framework and performance analysis	Compared prompting techniques (zero-shot, few-shot, CoT)	Improves requirement extraction accuracy and alignment
(Emebo et al., 2021)	How can common-sense knowledge and ontology help detect implicit FRs?	Text mining + ontology + common-sense reasoning	Requirements Engineering	Ontological corpora	Latent/implicit FRs	Conceptual validation	N/A	Improved detection of unspoken/implicit FRs

Reference	Main research question/problem	Used approach	Field Studied / Application domain	Dataset used	Attributes used for prediction	Evaluation of the approach	Comparison with other works	Result
(Sami, Waseem, et al., 2024)	How can AI agents prioritize requirements in agile workflows?	Web-based framework using MoSCoW, AHP + prompt engineering	Agile Requirements Engineering	Agile projects and JIRA-like tools	Priority weight, stakeholder needs	Implementation and use-case evaluation	N/A	Enabled integration with common PM systems like Trello
(Jia et al., 2024)	How can multi-agent LLM frameworks improve RE workflows?	Survey of 106 papers; proposes multi-agent decomposition	Requirements Engineering	N/A	RE tasks: elicitation, modeling, verification	Meta-analysis	N/A	Highlights shift from monolithic AI to collaborative agent systems
(Barnett et al., 2024)	What are the failure points in RAG systems used for RE?	Synthesis of 7 key failure points from RAG experience reports	Requirements Engineering, RAG	Real-world RAG deployment reports	Contextual window, hallucination, accuracy	Case synthesis and experiential lessons	N/A	Recommendations to improve RAG robustness in RE

Summary of Research Papers Based on Subject Domain Analysis Based On Text Mining Using a Large Language Model

The reviewed studies focused extensively on automating key aspects of requirements engineering, including classification (Alhoshan et al., 2023), specification (Krishna et al., 2024), domain modeling (Arulmohan et al., 2023), traceability (Guo et al., 2022), prioritization (Sami, Rasheed, et al., 2024), and completeness assessment (Luitel et al., 2024). These works aimed to address the inefficiencies of manual requirements extraction by leveraging NLP, ML, and especially LLMs. Several studies highlighted the integration of RAG and prompt engineering to improve contextual accuracy and reduce hallucinations in generated requirements (Lewis et al., 2020; Feng et al., 2024; Chen et al., 2023). A growing trend toward hybrid frameworks and multi-agent systems was also observed to enhance modularity and responsiveness in large-scale RE processes (Jia et al., 2024; Sami et al., 2024) .

Various AI methods were used, such as GPT-4, CodeLlama (Krishna et al., 2024), Bloom and LLaMA3 (Alhoshan et al., 2025), and domain-specific adaptations using frameworks like OPTIMA and UDAPDR (Guo et al., 2022; Saad-Falcon et al., 2023). Empirical studies included use cases in aerospace (Tikayat Ray et al., 2023), education (Krishna et al., 2024), and chemical domains (Wang et al., 2024), and explored both structured datasets (e.g., PROMISE NFR, PURE) and real-world textual documents. Evaluation methods ranged from human benchmarks (Sagodi et al., 2024) to static/dynamic verification (Liu & Bruel, 2024), with measured F1 scores and traceability checks used to benchmark performance (Das et al., 2024;Marques et al., 2024).

Findings across the literature consistently show that LLMs enhance the automation of FR extraction, domain comprehension, and specification generation, often matching or exceeding human baseline performance (Sagodi et al., 2024; Sridhara et al., 2023). Nonetheless, challenges such as hallucination control, compliance verification, and domain adaptation remain unresolved (Fan et al., 2023; Vogelsang, 2024). There is a clear demand for robust evaluation metrics, practical integration into development workflows, and ethical considerations. Despite these hurdles, the adoption of AI-enhanced RE tools, including RAG frameworks, formal modeling techniques, and ontology-driven validation, continues to reshape software engineering practices, setting a foundation for scalable and context-aware requirement automation (Raedler et al., 2023; Alharbi et al., 2024).

3. Proposed Methodology

3.1. Overview of the Proposed Methodology

The proposed method provides a structured model that systematizes the application of LLMs and RAG techniques in order to improve the consistency and compliance accuracy of FRs extraction from software documentation. This model solves gaps in automated requirements systems by integrating a modern language model approach to understanding frameworks with specific knowledge domain retrieval frameworks.

The methodology is outlined as a systematic approach that treats unstructured or semi-structured requirements documents as raw inputs and outputs polished, comprehensive, and uniform FR documents. This approach comprises document preprocessing, knowledge base construction, contextual information retrieval, intelligent requirements extraction, and quality evaluation phases. Each step has its own specific mechanisms tailored towards achieving certainty on the precision of the extracted requirements.

Employing LLMs with RAG fusion brings distinct benefits as compared to purely retrieval-based or generative approaches. The retrieval ensures that contextual content generation is aligned with the domain and organizational policy, while the LLM ensures sophisticated natural language understanding and generation of sophisticated features. Addressing common concerns put forth on pure generative approaches like imagination and coverage on pure retrieval-based methods.

3.2. Describing A Process Model Using BPMN Diagrams

In this section, using BPMN will demonstrate how the methodology is represented as a systematic workflow starting from data sources identification to validating the detailed requirements as an output of the process. Also, the model demonstrates strategically placed decision points, opportunities for parallel processing, as well as quality control steps within the flow of the methodology.

3.2.1. Workflow Commencement and Document Structuring

The BPMN diagram (Figure 1) illustrates a structured and systematic workflow for the automated extraction of FRs from various requirement documentations. This process utilizes the combination of LLMs and RAG methods to improve consistency and compliance. The first step in the methodology is accepting the gaps, which include different types of data sources as inputs to the system. Possible inputs include SRS documents, user stories, change requests, project backlogs, web pages, databases, interview notes, audio and video files, as well as compliance documents from different regulations. This process has an accommodating boundary that allows documents as inputs in the form of highly structured templates, unstructured narratives, or hybrid documents that are textual and/or visual in nature.

Document preprocessing, Knowledge base construction, Contextual retrieval, and Retrieval & generation via LLM steps are discussed in detail in the following section. Within the requirement specification creation steps, validated FRs are structured into formal documentation formats and stored in the specification repository. The evaluation framework presented in this research merges RAGAS-based metrics (Faithfulness, Answer Relevance, Recall@k) with specialized metrics designed for requirements engineering, including Technical Term Coverage and Compliance Score. Although RAGAS facilitates reference-free quality assessment for LLM outputs, the supplementary metrics guarantee conformity to industry-specific terminology, structural templates, and regulatory formats common in professional software requirements documents. The evaluation process is applied after each experimental run, enabling iterative improvements and early detection of issues such as hallucinations, domain misalignment, or template non-compliance. The evaluation results are logged and fed back into the prompt refinement loop if thresholds are not met.

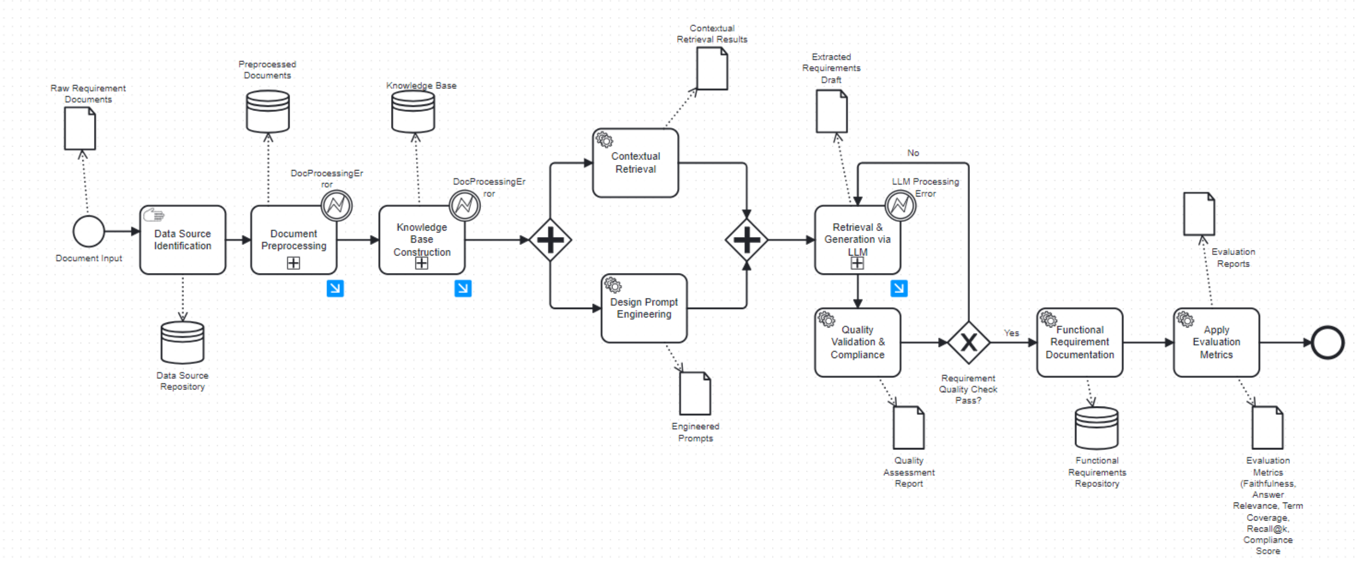

Raw Requirement Documents
Document Input
Data Source Identification
Data Source Repository
Preprocessed Documents
DocProcessingError
Document Preprocessing
Knowledge Base
DocProcessingError
Knowledge Base Construction
Contextual Retrieval Results
Contextual Retrieval
Design Prompt Engineering
Engineered Prompts
Extracted Requirements Draft
No
LLM Processing Error
Retrieval & Generation via LLM
Quality Validation & Compliance
Quality Assessment Report
Requirement Quality Check Pass?
Yes
Functional Requirement Documentation
Functional Requirements Repository
Evaluation Reports
Apply Evaluation Metrics
Evaluation Metrics (Faithfulness, Answer Relevance, Term Coverage, Recall@k, Compliance Score

Figure 1. BPMN Diagram for Proposed Methodology Main Process

3.2.2. Document Preprocessing

Once documents have been provided as input, the next step in the workflow is the Document Preprocessing Task, which is a collection of steps performed on the input documents. It will consist of the following tasks. Standardization of the document's format, extraction of text from certain document types, cleansing, which includes the removal of non-relevant content, capture, and cleaning of text, splitting of content into sections, as well as extraction of metadata, followed by classification. The operations performed in preprocessing make certain that optimized changes made in the subsequent steps will be smooth at any stage in the input pipeline while maintaining consistency in document structure.

A Quality Gate follows after the preprocessing phase to assess readiness for documents. Areas where the documents do not meet criteria are sent to a Manual Review Task, where specialists can fix missing details, formatting, or other systemic discrepancies before sending them back to the automated sequence. The documents that were successfully processed now proceed to the knowledge base construction step. The methodology encompasses comprehensive error recovery and handling strategies to ensure operational resilience when deployed in real-world contexts. Strategic error detection points are placed within the flow to monitor failure, quality loss, and resource constraints within a process stream and its associated system components. Document Processing Errors, such as file incompatibility or corruption, and information gaps, are managed by the Document Processing Error Handler. Recovery in this case would involve attempting the alternate format processing described within the document's structure, along with requesting, through manual intervention, permitting partial automation fallback to incomplete processing for cases that cannot be fully automated.

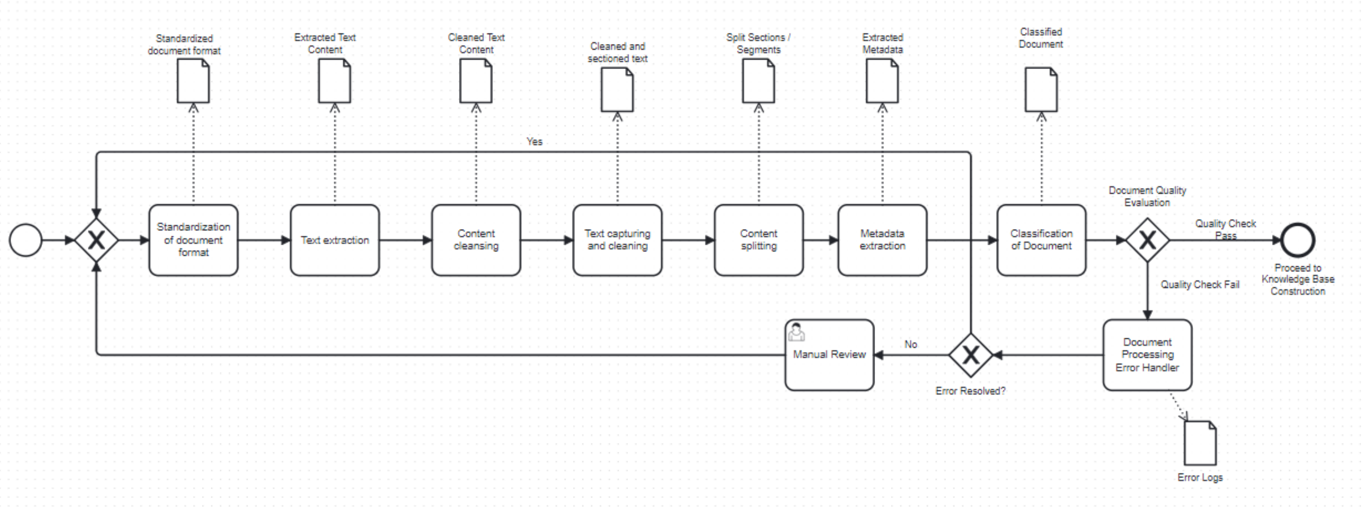

Standardized document format
Extracted Text Content
Cleaned Text Content
Cleaned and sectioned text
Split Sections / Segments
Extracted Metadata
Classified Document
Yes
Standardization of document format
Text extraction
Content cleansing
Text capturing and cleaning
Content splitting
Metadata extraction
Classification of Document
Document Quality Evaluation
Quality Check Pass
Proceed to Knowledge Base Construction
Quality Check Fail
Manual Review
No
Error Resolved?
Document Processing Error Handler
Error Logs

Figure 2. BPMN for Document Preprocessing Sub-Process

The methodology includes several critical decision points that control process flow order and quality to achieve the targets and objectives. These are implemented as exclusive gateways, which are route processing based on specific criteria and validation results. The Document Quality Gateway evaluates whether the input documents contain sufficient information to allow for automated document processing. Any documents that are missing essential information or contain significant structural problems are sent to manual preprocessing tasks, where human experts provide the necessary function before automation can continue.

3.2.3. Knowledge Base Construction and Enhancement

The Knowledge Base Construction Task is a pivotal element that differentiates this methodology from purely generative approaches. This sub-process is responsible for assembling a comprehensive, validated knowledge base that serves as the backbone for downstream RAG and LLM activities. Knowledge base construction synthesizes domain vocabulary, patterns, and entity relationships, industry standards, and expert retrospective analysis to ensure that the resulting knowledge base is both contextually rich and compliant with relevant standards. The knowledge base acts as a repository for information to be used in the retrieval part of the RAG architecture and as a source for the retrieval component of the RAG pipeline.

The sub-process begins by taking preprocessed documents as input and systematically analyzing them through parallel tracks. A Bag-of-Words model is applied to extract a comprehensive list of domain-specific terms, roles, and vocabulary. Concurrently, pattern and entity recognition techniques are used to identify and extract candidate entities and relationships. Following these initial analyses, the Bag-of-Words model construction and Graph construction for RAG are performed. The extracted vocabulary is formalized and stored within a Bag-of-Words repository. In parallel, using the entity and relation candidates, a Graph Knowledge Base is constructed where domain concepts become nodes and identified relationships become edges, thus supporting advanced semantic retrieval and graph-based reasoning in the later RAG phase.

The Standards Integration task incorporates relevant industry standards, organizational policies, and legal documents that govern the requirements specification process. Retrospective analysis further enhances the knowledge base with insights from past requirements specifications or expert input. All prior outputs are systematically merged and compiled into the main knowledge base during the Merge/Compile Knowledge Base task.

A Knowledge Validation Gateway confirms the completeness as well as the accuracy of the constructed knowledge base. Coverage assessment of the vocabulary, compliance with established standards, and consistency across diverse knowledge sources all check validation criteria. Knowledge bases that are validated proceed to the retrieval-augmented generation stage, whereas those that need enhancement go back to the construction task for further processing.

The problems that arise within knowledge construction and retrieval duties are handled by the Knowledge Base Error Handler. Recovery actions permitted are those involving fallback request, knowledge validation cancel request, and adaptive retrieval where primary knowledge sources are deficient or too incomplete and thus become unreachable.

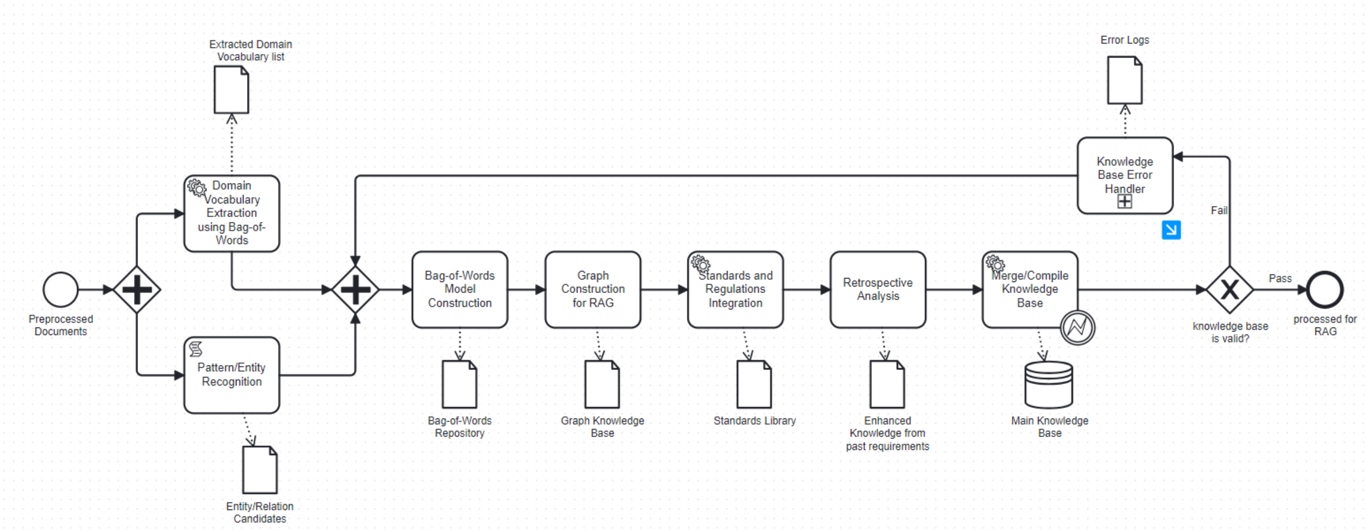

Extracted Domain Vocabulary list
Error Logs
Domain Vocabulary Extraction using Bag-of-Words
Knowledge Base Error Handler
Fail
Preprocessed Documents
Bag-of-Words Model Construction
Graph Construction for RAG
Standards and Regulations Integration
Retrospective Analysis
Merge/Compile Knowledge Base
Pass
knowledge base is valid?
processed for RAG
Pattern/Entity Recognition
Bag-of-Words Repository
Graph Knowledge Base
Standards Library
Enhanced Knowledge from past requirements
Main Knowledge Base
Entity/Relation Candidates

Figure 3. BPMN for Knowledge Base Construction Sub-Process

The Completeness Knowledge Base Gateway evaluates domain knowledge effectiveness in supporting retrieval-augmented generation. Incomplete knowledge bases initiate other knowledge-gathering tasks, including consulting experts, document analysis, and integration of external knowledge sources.

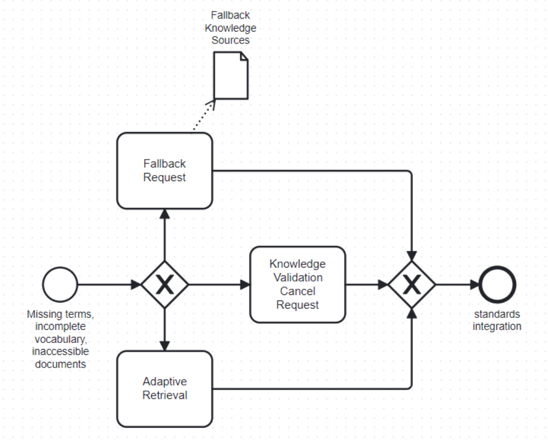

Fallback Knowledge Sources
Fallback Request
Knowledge Validation Cancel Request
Adaptive Retrieval
Missing terms, incomplete vocabulary, inaccessible documents
standards integration

Figure 4. BPMN for Knowledge Base Error Handler Sub-Process

3.2.4. Contextual Retrieval and Information Gathering

The Contextual Retrieval Task executes the retrieval part of the RAG architecture. This task performs semantic similarity matching between requirements and knowledge base entries, gathering information from relevant documents, pattern matching with pre-defined requirements templates, and compliance rule identification for applicable standards and regulations.

Advanced algorithms that focus on both the syntax and semantics of the knowledge base entry employ contextual retrieval requirements. Knowledge base embeddings are compared with dense vector representations of the requirements text to retrieve pertinent supporting information. Additional contextual factors such as project domain, stakeholder preferences, and technical constraints also shape the selection of relevant knowledge.

A parallel task of prompt engineering runs simultaneously with contextual retrieval to create tailored prompts for the language model component. This task integrates retrieved information into the model's prompts in a way that instructs the model to produce consistent and compliant outputs. The prompt engineering heuristic incorporates examples for few-shot learning, specifies constraints, outlines output format expectations, and defines quality thresholds.

3.2.5. Intelligent Requirements Generation and Processing

The LLM Processing Task constitutes the core generative functionality of the methodology. This task processes the provided prompts and contextual data to create structured FRs. The process incorporates natural language interpretation of the documents, integration of contextual information from the knowledge base, generation of structured requirements according to templates, and preliminary consistency checks against the provided exemplars.

The LLM performing the processing delineates several custom tasks that focus on distinct parts of requirements generation. In the Requirements Extraction task, an FR is identified and extracted from free text. In the Requirements Enhancement task, the identified requirements are made clearer and complete, and more specific. The Requirements Standardization task checks organizational template compliance with the document's structure and formatting. An Iterative Refinement Loop supports multiple processing passes for cases where the initial results do not meet the required pre-qualifying thresholds. This loop contains feedback mechanisms that mark certain quality concerns, providing directions to subsequent processing loops. The refinement processes stop when requirements of a specific generation are achieved against the preset defining criteria or the maximum iteration limits have been reached.

Issues of model unavailability, processing errors, or degradation in output quality are overseen by the LLM Processing Error Handler. Recovery in this case includes deployment of model alternatives, change requests through prompt adjustment, and fallback to manual processing for critical automatable requirements. The overarching error handling paradigm guarantees that the approach functions seamlessly across different institutional settings, as it ensures scholarly benchmarks when specific processing parts fail. This feature is critical for actual use in industrial requirements engineering applications where dependability and uniformity are crucial.

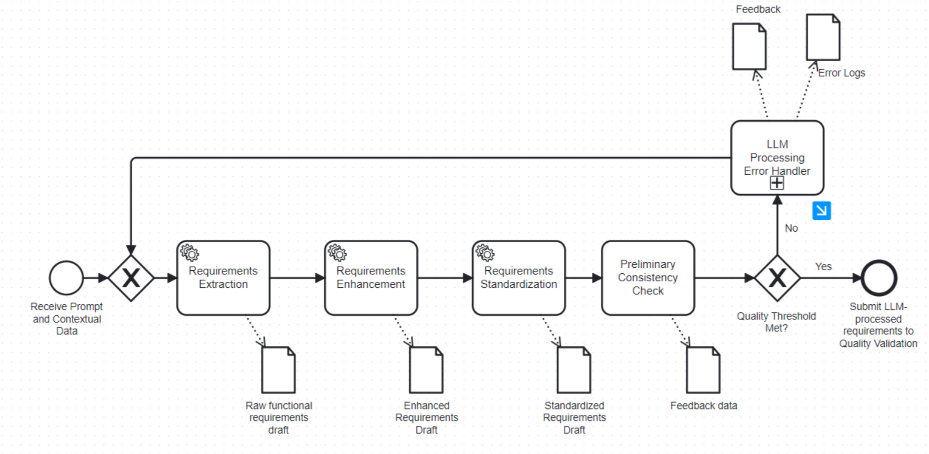

Feedback
Error Logs
LLM Processing Error Handler
No
Yes
Receive Prompt and Contextual Data
Requirements Extraction
Requirements Enhancement
Requirements Standardization
Preliminary Consistency Check
Quality Threshold Met?
Submit LLM-processed requirements to Quality Validation
Raw functional requirements draft
Enhanced Requirements Draft
Standardized Requirements Draft
Feedback data

Figure 5. BPMN for LLM Processing Sub-Process

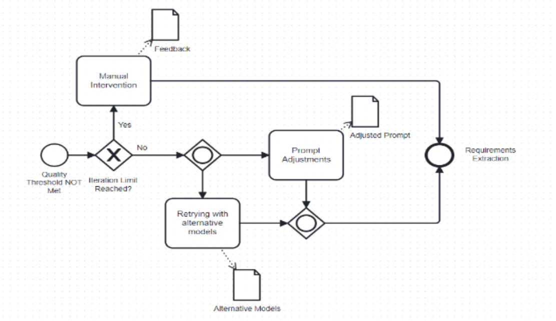


Figure 6. BPMN for LLM Processing Error Handler

3.2.6. Quality Validation and Compliance Verification

The Quality Validation Task undertakes the design and implementation of system assessment processes aimed at ensuring all generated requirements adhere to the appropriate predetermined benchmarks. This task defines consistency checking across generated requirements. Compliance verification against designated benchmarks and legislation, completeness assessment for potential functional area absence, and clarity assessment to guarantee the absence of vague, unambiguous, and hypothesisable wording within the requirements.

Both automated checking mechanisms and structured review protocols are employed in the validation processes. Mechanized checks are capable of detecting relevant quality problems such as the use of vague language, the absence of relevant parts, and contradictory terminologies. Structured review protocols help direct specialists to evaluate the generated requirements when automated checking becomes inadequate.

A Decision Gateway determines the overall requirements generated by adequate threshold targets, as to which the quality target pass is achieved for the final output. Requirements that pass validation undergo Output Formatting Tasks, while those that need improvement revert to prior processing stages based on the specific quality concerns flagged. This feedback loop facilitates and sustains perpetual enhancement of requirements quality.

3.2.7. Process Flow Control and Decision Points

The Requirements Quality Gateway represents the most critical decision posture in methodology. This gateway scrutinizes the generated requirements against a comprehensive quality checklist that includes clarity, completeness, consistency, and compliance. Requirements not meeting the predetermined quality thresholds are routed back to the designated corrective processing stage based on the highlighted deficiencies.

3.2.8. Opportunities for Parallel Processing and Optimization

The method contains multiple gaps where parallel processing can be implemented to optimize efficiency and reduce overall processing time. Contextual retrieval as well as prompt engineering tasks function in parallel as a means to improve processing efficiency. While relevant knowledge is retrieved from the knowledge base, tailored prompts are prepared to maximize performance from the LLM simultaneously. This approach improves overall latency without compromising quality.

Concurrent processing of multiple requirements documents is possible when system resources are available, enabling batch processing for extensive-scale requirements engineering projects. The methodology utilizes parallel processing streams, which maintain a consistent level of quality balanced by applied workload distribution management of concurrent resource streams.

4. Results Measurement Metrics

The *RAGAS* framework (Retrieval Augmented Generation Assessment) offers a reference-free evaluation methodology for RAG-based systems, which is highly relevant to this thesis involving FR extraction using LLM + RAG (Es et al., 2024). It enables multi-dimensional assessment without requiring human-annotated ground truths, and it provides metrics that measure different aspects of output quality, such as faithfulness, relevance, and grounding, without requiring human-annotated ground truth. Given the goal of this thesis to improve the consistency and compliance of automatically generated FRs, RAGAS serves as an ideal basis for selecting evaluation metrics that are both automated and interpretable.

A total of five quantitative metrics have been identified for this study, derived from RAGAS principles, domain-specific adaptations, and standard in requirements engineering to evaluate the efficacy of the suggested LLM + RAG-driven framework for extracting FR. These metrics were chosen to thoroughly assess the relevance, precision, alignment with the domain, adherence to structure, and quality of retrieval of the produced requirements. The assessment utilizes not just reference annotations but also incorporates both automations guided by LLM and constraints specific to the domain.

Faithfulness

This metric measures whether the generated requirement is grounded in the retrieved context. The purpose is to prevent hallucinations and ensure traceability to original documents. This process can begin with segmenting each generated FR into a set of clauses. So, each clause will be checked against the retrieved or original source document using a verification function. Like that, this metric ensures that the extracted requirements are not only contextually aligned but also traceable to the original documentation.

Answer Relevance

Evaluates how well the generated requirement responds to the original prompt or used need. Or the system needs. This ensured semantic alignment with input queries.

Technical Term Coverage

Measures how many domain-specific terms are used in the requirement. This will ensure domain relevance and terminology precision. This evaluation can begin with constructing a terminology vocabulary, collected from domain-specific sources (Standard documents, Technical specifications, or expert-verified glossaries).

Recall@k

Measures whether expected requirements appear in the top k generated outputs. This will evaluate recall performance where annotated datasets exist. This will ensure that the knowledge fed into the generative model is not only relevant but sufficient to support correct requirement generation.

Compliance Score

Compliance score checks whether the generated FRs adhere to structural, linguistic, and regulatory norms. This framework incorporates three primary dimensions such as alignment to an organizational or industry-approved requirement template, alignment with standards such as IEEE 830 (Defines how to write a well-structured SRS document) or ISO/IEC/IEEE 29148 (Covers both system and SRs, FRs and NFRs), and satisfaction of domain-specific regulatory criteria.

5. Initial Experiment

5.1. Experimental Domain Selection and Justification

To evaluate the performance and applicability of the proposed hybrid LLM + RAG-based methodology for FR extraction, the domain selected for experimentation is **Enterprise-Scale Agile Software Development in the Healthcare Sector**, particularly under strict regulatory constraints. This domain was selected for the following reasons:

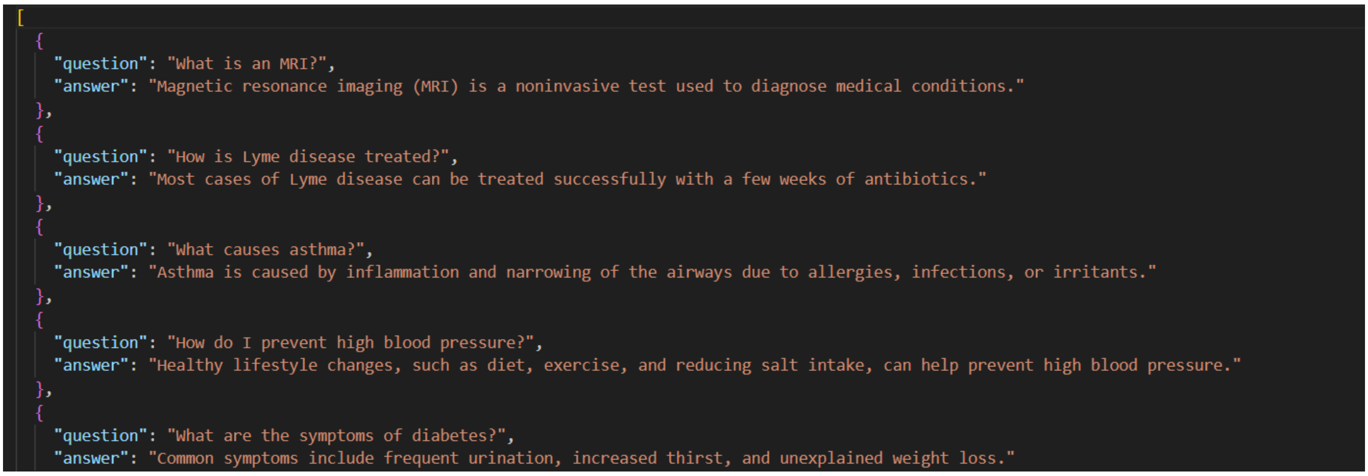
Heterogeneous and Multi-Format Input Sources: Modern enterprise-level healthcare software projects are characterized by complex, agile development processes. These generate heterogeneous documentation, including Agile artifacts such as user stories, change requests, and backlog items, Regulatory compliance documents (HIPAA, HL7, FDA), SRS documents, API schemas, and patient-facing web content, Informal or semi-structured sources such as interview notes, meeting transcripts, and chat logs. This diversity creates an optimal setting for testing the robustness of the pipeline across various types of inputs, especially during the document preprocessing and contextual retrieval stages.

High Compliance Demands and Risk Sensitivity: Healthcare systems operate in one of the most compliance-driven environments, with regulations such as: HIPAA: Ensures data privacy and security for patient health records, HL7/FHIR: Mandates interoperability standards for data exchange, FDA 21 CFR Part 11: Governs electronic records and digital signatures for traceability and integrity**.** These regulatory limitations directly influence and define FRs. As a result, this area is particularly appropriate for assessing the Compliance Score and Faithfulness metrics within the methodology. It also tests the model's ability to prevent hallucinations and generate outputs that are both structurally and semantically correct.

Industrial Relevance and Real-World Impact: Enterprise-scale healthcare software is not only highly complex but also directly affects patient safety, system reliability, and legal liability. Automating and standardizing the extraction of FRs in such settings offers realistic use cases for LLM-based solutions**,** high practical relevance for industry adoption**,** significant value in reducing manual errors and ensuring traceability**.** This industrial realism increases the external validity of the thesis results and opens the path for future technology transfer and productization of the proposed approach.

5.2. Feasibility Analysis of Proposed Methodology

To evaluate the technical viability of the proposed methodology, an initial experiment was conducted using the MedQuAD dataset a domain-specific corpus of medical question-answer (QA) pairs. The objective was to assess whether the integration of RAG with LLMs could effectively extract FRs from semi-structured textual inputs.



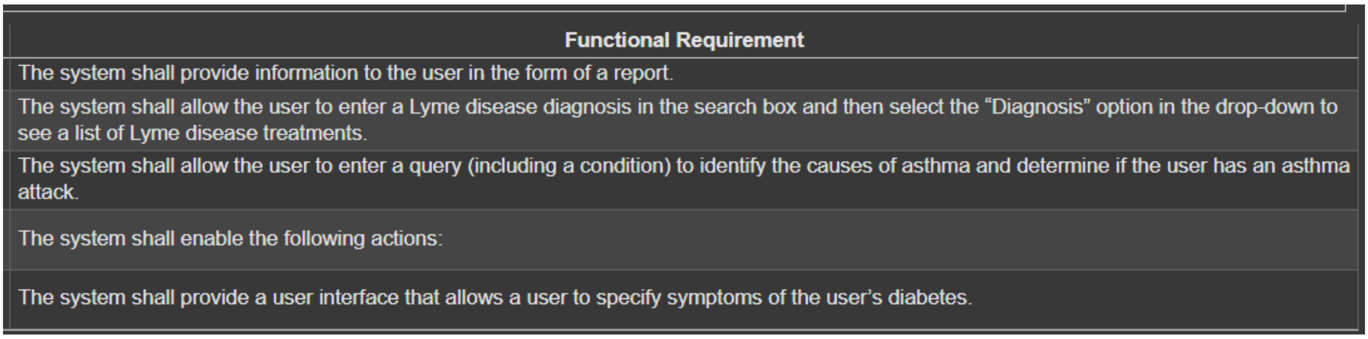
```
[
  {
    "question": "What is an MRI?",
    "answer": "Magnetic resonance imaging (MRI) is a noninvasive test used to diagnose medical conditions."
  },
  {
    "question": "How is Lyme disease treated?",
    "answer": "Most cases of Lyme disease can be treated successfully with a few weeks of antibiotics."
  },
  {
    "question": "What causes asthma?",
    "answer": "Asthma is caused by inflammation and narrowing of the airways due to allergies, infections, or irritants."
  },
  {
    "question": "How do I prevent high blood pressure?",
    "answer": "Healthy lifestyle changes, such as diet, exercise, and reducing salt intake, can help prevent high blood pressure."
  },
  {
    "question": "What are the symptoms of diabetes?",
    "answer": "Common symptoms include frequent urination, increased thirst, and unexplained weight loss."
```

Figure 7. MedQuAd DataSet Sample

MedQuAD is a publicly available dataset created by the U.S. National Library of Medicine, comprising approximately 47,000 question-answer (QA) pairs across 12 medical subdomains. These QA entries are derived from trusted, expert-curated sources. In this study, only 50 QA pairs were taken and treated as domain-specific knowledge units from which FRs could be extracted. By leveraging this medically oriented dataset, the experiment simulated a real-world scenario of extracting requirements from domain-focused documentation, helping validate the contextual robustness and semantic accuracy of the proposed methodology.

The experimental pipeline consisted of preprocessing QA entries, constructing a semantic retrieval index using FAISS with sentence-transformer embeddings, and generating FRs using OpenAI GPT-based LLMs with tailored prompts.

5.3. Results Obtained from the Initial Experiment



Functional Requirement
The system shall provide information to the user in the form of a report.
The system shall allow the user to enter a Lyme disease diagnosis in the search box and then select the "Diagnosis" option in the drop-down to see a list of Lyme disease treatments.
The system shall allow the user to enter a query (including a condition) to identify the causes of asthma and determine if the user has an asthma attack.
The system shall enable the following actions:
The system shall provide a user interface that allows a user to specify symptoms of the user's diabetes.

Figure 8. Extracted FRs from the Initial Experiment

Even with a simplified knowledge base and a limited scope of dataset, the outputs produced were well-grounded in context, structured appropriately (e.g., adhering to "The system shall…" formats), and semantically consistent with the relevant domain. According to the outputs generated, the majority of the FRs adhered to the established "The system shall…" structure, demonstrating strong compliance with formal conventions of FR writing. Most of the FRs appeared to be accurately aligned with the input question-and-answer content, indicating solid contextual grounding. Certain terminology and intent were preserved effectively (e.g., "Lyme disease," "asthma attack"), affirming the understanding of the domain. However, one result indicated partial completion, revealing a shortcoming in the output's completeness, which may necessitate the implementation of iterative refinement logic or threshold-based validation in future phases.

These results demonstrate the core feasibility of the proposed approach. Specifically, the experiment confirmed that combining retrieval mechanisms with LLM-based generation enables accurate and consistent extraction of FRs, even in a specialized domain like healthcare. This experiment thus serves as an early validation of the technical soundness of the proposed methodology. It provides confidence in scaling the approach to more complex datasets and incorporating additional components such as structured knowledge bases, formal compliance checks, and automated evaluation metrics.

6. Conclusions

Based on the performed comprehensive review of existing NLP, LLM, and RAG methodologies, the results indicate that although LLMs like GPT-4 significantly enhance the automation of extracting FRs, challenges such as hallucinations and adherence to domain-specific standards persist. Analyzing related studies and existing approaches reveals a gradual transition from conventional NLP to integrated hybrid AI approaches that merge retrieval and generation to boost accuracy. The review demonstrated that fully automated, consistent, and compliance-aware requirements extraction is still an open research gap.

Based on the performed design of a structured hybrid framework integrating LLMs with RAG, the obtained result shows that combining knowledge base retrieval with advanced natural language generation improves the precision and compliance of extracted FRs. Analysis of related workflows and BPMN process modeling shows the methodology provides clear error handling, iterative refinement, and quality validation mechanisms to ensure reliable outputs. The experiment performed demonstrated that parallel processing and decision gateways enable scalability and operational robustness in industrial settings.

The proposed methodology presents a systematic and hybrid approach that integrates LLMs with RAG to automate FRs extraction. It combines advanced natural language understanding with contextual retrieval from a comprehensive knowledge base to improve accuracy, consistency, and compliance. The approach incorporates iterative refinement, quality validation, and error management systems, rendering it scalable and flexible for various software documentation types and practical requirements engineering processes.

The established evaluation metrics offer an organized and automated way to evaluate the quality, relevance, and adherence of FRs derived using the proposed LLM + RAG framework. By utilizing both domain-specific limitations and reference-free scoring methods, these metrics facilitate thorough validation of the generated outputs without depending on human-annotated references. This guarantees that the method is scalable, interpretable, and appropriate for practical application in software engineering settings where consistency and traceability are essential.

The preliminary experiment, carried out using the MedQuAD dataset, effectively showcased the viability of combining RAG with LLMs for the extraction of FRs. The pipeline generated requirements that were contextually precise and structurally coherent, thereby validating the essential components of the proposed methodology. This initial evaluation affirms the technical reliability of the approach and lays a robust groundwork for expanding the system to accommodate more intricate and diverse data sources in future stages of the research.

7. References

Ahmad, K., Abdelrazek, M., Arora, C., Bano, M., & Grundy, J. (2023). Requirements engineering for artificial intelligence systems: A systematic mapping study. *Information and Software Technology*, *158*, 1–45. https://doi.org/10.1016/j.infsof.2023.107176

Aishwarya, V. (2023). A Prompt Engineering Approach for Structured Data Extraction from Unstructured Text Using Conversational LLMs. *ACM International Conference Proceeding Series*, 183–189. https://doi.org/10.1145/3639631.3639663

Alharbi, R., Tamma, V., Grasso, F., & Payne, T. (2024). An Experiment in Retrofitting Competency Questions for Existing Ontologies. In *Proceedings of the ACM Symposium on Applied Computing* (pp. 1650–1658). https://doi.org/10.1145/3605098.3636053

Alhoshan, W., Ferrari, A., & Zhao, L. (2023). Zero-shot learning for requirements classification: An exploratory study. *Information and Software Technology*, *159*. https://doi.org/10.1016/j.infsof.2023.107202

Alhoshan, W., Ferrari, A., & Zhao, L. (2025). How Effective are Generative Large Language Models in Performing Requirements Classification ? *ACM Transactions on Software Engineering and Methodology*, *00*(00). https://doi.org/https://doi.org/10.48550/arXiv.2504.16768

Arora, C., Grundy, J., & Abdelrazek, M. (2024). Advancing Requirements Engineering Through Generative AI: Assessing the Role of LLMs. In *Generative AI for Effective Software Development* (pp. 129–148). https://doi.org/10.1007/978-3-031-55642-5_6

Arora, C., Herda, T., & Homm, V. (2024). Generating Test Scenarios from NL Requirements Using Retrieval-Augmented LLMs: An Industrial Study. In *Proceedings of the IEEE International Conference on Requirements Engineering* (pp. 240–251). https://doi.org/10.1109/RE59067.2024.00031

Arulmohan, S., Meurs, M. J., & Mosser, S. (2023). Extracting Domain Models from Textual Requirements in the Era of Large Language Models. *Proceedings - 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion, MODELS-C 2023*, 580–587. https://doi.org/10.1109/MODELS-C59198.2023.00096

Arvidsson, S., & Axell, J. (2023). *Prompt engineering guidelines for LLMs in Requirements Engineering*. https://gupea.ub.gu.se/handle/2077/77967

Barnett, S., Kurniawan, S., Thudumu, S., Brannelly, Z., & Abdelrazek, M. (2024). Seven failure points when engineering a retrieval augmented generation system. *Proceedings - 2024 IEEE/ACM 3rd*

International Conference on AI Engineering - Software Engineering for AI, CAIN 2024, *1*(1), 194–199. https://doi.org/10.1145/3644815.3644945

Belzner, L., Gabor, T., & Wirsing, M. (2024). Large Language Model Assisted Software Engineering: Prospects, Challenges, and a Case Study. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *14380 LNCS*, 355–374. https://doi.org/10.1007/978-3-031-46002-9_23

Chen, B., Zhang, Z., Langrené, N., & Zhu, S. (2023). *Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review*. 1–58. https://doi.org/10.1016/j.patter.2025.101260

Das, S., Deb, N., Cortesi, A., & Chaki, N. (2024). Extracting goal models from natural language requirement specifications. *Journal of Systems and Software*, *211*(January). https://doi.org/10.1016/j.jss.2024.111981

Emebo, O., Varde, A. S., & Daramola, O. (2021). *Common Sense Knowledge, Ontology and Text Mining for Implicit Requirements*. https://doi.org/https://doi.org/10.48550/arXiv.2103.11302

Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAS: Automated Evaluation of Retrieval Augmented Generation. *EACL 2024 - 18th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of System Demonstrations*, 150–158.

Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S., & Zhang, J. M. (2023). Large Language Models for Software Engineering: Survey and Open Problems. *Proceedings - 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering, ICSE-FoSE 2023*, 31–53. https://doi.org/10.1109/ICSE-FoSE59343.2023.00008

Feng, N., Marsso, L., Yaman, S. G., Standen, I., Baatartogtokh, Y., Ayad, R., De Mello, V. O., Townsend, B., Bartels, H., Cavalcanti, A., Calinescu, R., & Chechik, M. (2024). Normative Requirements Operationalization with Large Language Models. *Proceedings of the IEEE International Conference on Requirements Engineering*, 129–141. https://doi.org/10.1109/RE59067.2024.00022

Guo, X., Li, B., & Yu, H. (2022). Improving the Sample Efficiency of Prompt Tuning with Domain Adaptation. *Findings of the Association for Computational Linguistics: EMNLP 2022*, 3523–3537. https://doi.org/10.18653/v1/2022.findings-emnlp.258

Jia, J., Zhang, Y., Zhang, Y., Liu, J., Runwal, B., Diffenderfer, J., Kailkhura, B., & Liu, S. (2024). *SOUL: Unlocking the Power of Second-Order Optimization for LLM Unlearning*. 4276–4292. http://arxiv.org/abs/2404.18239

Krishna, M., Gaur, B., Verma, A., & Jalote, P. (2024a). Using LLMs in Software Requirements Specifications: An Empirical Evaluation. In *Proceedings of the IEEE International Conference on Requirements Engineering* (pp. 475–483). https://doi.org/10.1109/RE59067.2024.00056

Krishna, M., Gaur, B., Verma, A., & Jalote, P. (2024b). Using LLMs in Software Requirements Specifications: An Empirical Evaluation. *Proceedings of the IEEE International Conference on Requirements Engineering*, 475–483. https://doi.org/10.1109/RE59067.2024.00056

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W. T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, *2020-Decem*.

Liu, Y., & Bruel, J. M. (2024). Modeling and Verification of Natural Language Requirements based on States and Modes. *Formal Aspects of Computing*, *36*(2). https://doi.org/10.1145/3640822

Liuska, J. (2024). *Bachelor's Thesis- ENHANCING LARGE LANGUAGE MODELS FOR DATA ANALYTICS THROUGH DOMAIN-SPECIFIC CONTEXT CREATION*.

Luitel, D., Hassani, S., & Sabetzadeh, M. (2024). Improving requirements completeness: automated assistance through large language models. *Requirements Engineering*, *29*(1), 73–95. https://doi.org/10.1007/s00766-024-00416-3

Malan, R., & Bredemeyer, D. (2001). Functional Requirements and Use Cases. In *White Paper* (Issue 8/3/01, pp. 1–10).

Marques, N., Silva, R. R., & Bernardino, J. (2024). Using ChatGPT in Software Requirements Engineering: A Comprehensive Review. *Future Internet*, *16*(6), 1–21. https://doi.org/10.3390/fi16060180

Raedler, S., Berardinelli, L., Winter, K., Rahimi, A., & Rinderle-Ma, S. (2023). Bridging MDE and AI: A Systematic Review of Domain-Specific Languages and Model-Driven Practices in AI Software Systems Engineering. *Software and Systems Modeling*. https://doi.org/10.1007/s10270-024-01211-y

Saad-Falcon, J., Khattab, O., Santhanam, K., Florian, R., Franz, M., Roukos, S., Sil, A., Sultan, M. A., & Potts, C. (2023). UDAPDR: Unsupervised Domain Adaptation via LLM Prompting and Distillation of Rerankers. *EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings*, 11265–11279. https://doi.org/10.18653/v1/2023.emnlp-main.693

Sagodi, Z., Siket, I., & Ferenc, R. (2024). Methodology for Code Synthesis Evaluation of LLMs Presented by a Case Study of ChatGPT and Copilot. *IEEE Access*, *12*, 72303–72316. https://doi.org/10.1109/ACCESS.2024.3403858

Sami, M. A., Rasheed, Z., Waseem, M., Zhang, Z., Herda, T., & Abrahamsson, P. (2024). *Prioritizing Software Requirements Using Large Language Models*. 1–11. https://doi.org/https://doi.org/10.48550/arXiv.2103.11302

Sami, M. A., Waseem, M., Zhang, Z., Rasheed, Z., Systä, K., & Abrahamsson, P. (2024). *AI based Multiagent Approach for Requirements Elicitation and Analysis*. https://doi.org/https://doi.org/10.48550/arXiv.2409.00038

Sonbol, R., Rebdawi, G., & Ghneim, N. (2022). The Use of NLP-Based Text Representation Techniques to Support Requirement Engineering Tasks: A Systematic Mapping Review. In *IEEE Access* (Vol. 10, pp. 62811–62830). https://doi.org/10.1109/ACCESS.2022.3182372

Sridhara, G., G., R. H., & Mazumdar, S. (2023). *ChatGPT: A Study on its Utility for Ubiquitous Software Engineering Tasks*. http://arxiv.org/abs/2305.16837

Tikayat Ray, A., Cole, B. F., Pinon Fischer, O. J., Bhat, A. P., White, R. T., & Mavris, D. N. (2023). Agile Methodology for the Standardization of Engineering Requirements Using Large Language Models. *Systems*, *11*(7), 1–28. https://doi.org/10.3390/systems11070352

Umar, M. A., & Lano, K. (2024). Advances in automated support for requirements engineering: a systematic literature review. *Requirements Engineering*, *29*(2), 177–207. https://doi.org/10.1007/s00766-023-00411-0

Vogelsang, A. (2024a). From Specifications to Prompts: On the Future of Generative Large Language Models in Requirements Engineering. In *IEEE Software* (Vol. 41, Issue 5, pp. 9–13). https://doi.org/10.1109/MS.2024.3410712

Vogelsang, A. (2024b). From Specifications to Prompts: On the Future of Generative Large Language Models in Requirements Engineering. *IEEE Software*, *41*(5), 9–13. https://doi.org/10.1109/MS.2024.3410712

Vogelsang, A., & Fischbach, J. (2024). *Using Large Language Models for Natural Language Processing Tasks in Requirements Engineering: A Systematic Guideline*. 1–22. http://arxiv.org/abs/2402.13823

Wang, Q., Zhang, Z., Li, H., Liu, X., Han, J., Zhao, H., & Ji, H. (2024). Chem-FINESE: Validating Fine-Grained Few-shot Entity Extraction through Text Reconstruction. *EACL 2024 - 18th Conference of the European Chapter of the Association for Computational Linguistics, Findings of EACL 2024*, 1–16.

Wei, B. (2024). Requirements are All You Need: From Requirements to Code with LLMs. *Proceedings of the IEEE International Conference on Requirements Engineering*, 416–422. https://doi.org/10.1109/RE59067.2024.00049

Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M. A., Chioasca, E.-V., & Batista-Navarro, R. T. (2020). *Natural Language Processing (NLP) for Requirements Engineering: A Systematic Mapping Study*. *v*. https://doi.org/https://doi.org/10.48550/arXiv.2004.01099