Mälardalen University Licentiate Thesis 378

AI-AUGMENTED REQUIREMENTS ENGINEERING FOR INDUSTRIAL SYSTEMS

Sarmad Bashir

Mälardalen University

Mälardalen University Press Licentiate Theses
No. 378

AI-AUGMENTED REQUIREMENTS ENGINEERING FOR INDUSTRIAL SYSTEMS

Sarmad Bashir

2025

School of Innovation, Design and Engineering

ISBN 978-91-7485-720-7
ISSN 1651-9256
Printed by E-Print AB, Stockholm, Sweden

Abstract

Engineering large-scale industrial systems requires an efficient Requirements Engineering (RE) process to manage the complexity resulting from continuous technological advancements. In manufacturing domains such as railways, the complexity of software-intensive systems is growing due to evolving standards, infrastructure specifications, and increasing customer expectations. Typically, the RE process begins with analyzing extensive tender documents from external customers to assess project feasibility. This analysis is critical, as the tender documents define the scope and the standards to which the system-to-be must comply. Once validated and agreed upon, the requirements are distributed among various subsystem teams for development and testing. During implementation, the evolving requirements are cross-referenced with existing technical documents to ensure consistency across project artifacts and prevent integration issues within subsystems. However, the reliance on manual efforts in performing these RE tasks makes the process labor-intensive and time-consuming, often leading to project scope creep in industrial settings.

This thesis empirically investigates Artificial Intelligence (AI), particularly Large Language Models (LLMs)-based solutions, to augment the RE process for realizing complex industrial systems. The proposed solutions aim to provide decision support to reduce the manual efforts typically required for (i) identifying requirements from other supporting information in tender documents, (ii) detecting ambiguous requirements and explaining them, (iii) allocating validated requirements to appropriate subsystem teams for development and (iv) addressing requirement-related queries during the development and release phases of the project. Consequently, this research contributes to enhancing requirements management in complex industrial systems by enabling more efficient and informed decision-making.

Sammanfattning

Att konstruera storskaliga industriella system kräver en effektiv Requirements Engineering (RE)-process för att hantera den komplexitet som följer av kontinuerliga tekniska framsteg. Inom tillverkningsdomäner som järnväg ökar komplexiteten i mjukvaruintensiva system på grund av standarder som förändras, infrastrukturspecifikationer och ökade kundförväntningar. RE-processen inleds vanligen med att omfattande anbudsunderlag från externa beställare analyseras för att bedöma projektets genomförbarhet. Denna analys är avgörande eftersom anbudsunderlaget fastställer omfattningen och de standarder som det framtida systemet måste uppfylla. När kraven har validerats och godkänts fördelas de till olika delsystemteam för utveckling och test. Under implementeringen korsrefereras de föränderliga kraven med befintlig teknisk dokumentation för att säkerställa konsekvens emellan artefakter och förhindra integrationsproblem mellan delsystem. Det manuella arbete som krävs för dessa RE-uppgifter gör dock processen arbets- och tidskrävande, vilket ofta leder till 'scope creep' i industriella projekt.

Denna avhandling undersöker empiriskt AI-, särskilt LLMs-baserade, lösningar för att stärka RE-processen vid realisering av komplexa industriella system. De föreslagna lösningarna syftar till att ge beslutsstöd för att minska det manuella arbete som normalt krävs för (i) att identifiera krav bland övrig information i anbudsunderlag, (ii) att upptäcka tvetydiga krav och förklara dem, (iii) att fördela validerade krav till lämpliga delsystemteam för utveckling och (iv) att besvara kravsrelaterade frågor under projektets utvecklings- och releasefaser. Därmed bidrar forskningen till förbättrad kravhantering i komplexa industriella system genom att möjliggöra effektivare och mer välgrundade beslut.

To my Family

Acknowledgments

This licentiate thesis marks the halfway point of my doctoral journey and has been a valuable learning experience both professionally and personally. Along the way, I have met many wonderful people who have made this journey better and more meaningful.

First, I would like to thank my supervisors (Markus Bohlin, Mehrdad Saadatmand and Eduard Paul Enoiu) for giving me the freedom to explore research ideas, for providing opportunities to grow, and for their constructive feedback throughout this work.

Special thanks to my PhD colleagues at RISE (Abbas, Samaneh, Alireza) and MDU (Johan, Karim) for their discussions and for making this journey more enjoyable. I am also grateful to Per Erik Strandberg from Westermo and Zulqarnain Haider from Alstom for being the best industrial champions one could wish for, and for providing the support and collaboration that have been essential to this work.

Finally, I am deeply grateful to my family, my brother Sannan, for his constant support and encouragement. My wife, Nawal, for always keeping me grounded, and our wonderful boy, Yousef, who brings joy to our lives every day.

The work presented in this thesis has been partially funded by the KKS INDTECH industrial school (Grant No. 20200132) and by the SmartDelta (ITEA3), AIDOaRt (KDT Grant No. 101007350), and MATISSE (KDT Grant No. 101140216) projects.

Sarmad Bashir, Västerås, August 2025

List of Publications

Papers included in this thesis[1]

Paper A: *Sarmad Bashir*, Muhammad Abbas, Mehrdad Saadatmand, Eduard Paul Enoiu, Markus Bohlin, Pernilla Lindberg. "*Requirement or Not, That is the Question: A Case from the Railway Industry*". In the 29th International Working Conference on Requirement Engineering: Foundation for Software Quality (REFSQ 2023).

Paper B: *Sarmad Bashir*, Alessio Ferrari, Abbas Khan, Per Erik Strandberg, Zulqarnain Haider, Mehrdad Saadatmand, Markus Bohlin."*Requirements Ambiguity Detection and Explanation with LLMs: An Industrial Study*". In the 41st International Conference on Software Maintenance and Evolution (ICSME 2025).

Paper C: *Sarmad Bashir*, Muhammad Abbas, Alessio Ferrari, Mehrdad Saadatmand, Pernilla Lindberg. "*Requirements Classification for Smart Allocation: A Case Study in the Railway Industry*". In the 31st International Requirements Engineering Conference (RE 2023).

Paper D: Md Saleh Ibtasham, *Sarmad Bashir*, Muhammad Abbas, Zulqarnain Haider, Mehrdad Saadatmand, Antonio Cicchetti. "*ReqRAG: Enhancing Software Release Management through Retrieval-Augmented LLMs: An Industrial Study*". In the 31st International Working Conference on Requirement Engineering: Foundation for Software Quality (REFSQ 2025).

[1] The included publications have been reformatted to comply with the thesis layout.

Related publications, not included in this thesis

Paper E: Muhammad Abbas, *Sarmad Bashir*, Mehrdad Saadatmand, Eduard Paul Enoiu, Daniel Sundmark. "*Requirements Similarity and Retrieval*". In Ferrari, A., Ginde, G. (eds), Handbook on Natural Language Processing for Requirements Engineering. Springer, Cham (2025).

Paper F: *Sarmad Bashir*. "*Towards AI-centric Requirements Engineering for Industrial Systems*". In the Doctoral Symposium of IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings (ICSE 2024).

Contents

I

Thesis

Chapter 1

Introduction

The development of large-scale industrial systems is a complex and multi-disciplinary process that requires coordinated efforts for its realization. The recent digitalization trend in the manufacturing industry, such as railways, is fundamentally altering system development practices [1, 2]. This is due to the increasing complexity of modern software-intensive systems, driven by the growing demand for technological advancements to enhance efficiency. In this evolving landscape, Requirements Engineering (RE) becomes essential for effectively managing the growing complexity and addressing the evolving needs of customers throughout the system development lifecycle. This is evident in the railway industry, where RE establishes a foundation to ensure reliability by documenting clear, concise, and testable requirements of the system under development [3].

In this context, the RE process typically begins with the specification of high-level customer requirements from tender documents, followed by a feasibility analysis involving iterative clarification with stakeholders to resolve ambiguities and ensure validation. Once the requirements are validated and agreed upon, they are allocated to various dedicated subsystems teams, which include, but are not limited to, traction, propulsion control, doors, and brakes for development and testing [4]. Subsequently, these requirements are managed through the release management process, where engineers cross-reference technical documents to ensure that all changes are accurately reflected and aligned with system-level requirements and regulatory standards [5].

While current RE practices have been effective in developing highly reliable systems in the railway industry, there is a growing imperative for enhanced support in the process [6]. As system complexity increases, traditional requirements management becomes increasingly inadequate, leading to prolonged feedback loops and significant delays in system development and delivery. Furthermore, the continuous reliance on manual practices requires engineers to dedicate considerable time and effort to perform the RE tasks, making requirements management across the system development lifecycle both tedious and time-intensive.

Over the years, Artificial Intelligence (AI), particularly Natural Language Processing (NLP) techniques, has been increasingly employed to augment the RE process, including tasks like classifying functional and non-functional requirements, identifying linguistic issues, and establishing traceability links between requirements [7, 8, 9]. Notably, this is due to empirical evidence showing that natural language (NL) remains the standard notation for specifying requirements in the industry [10]. As a result, NLP techniques based on language models present considerable potential to support RE practitioners in carrying out various tasks on textual requirements documents. On the other hand, while proposed approaches have shown potential in automating and enhancing RE tasks, most existing techniques do not scale effectively in industrial contexts, mainly due to their limited ability to capture domain-specific knowledge [11]. Furthermore, the reliability of automated solutions in such settings remains a concern, as their outputs often lack adequate explanations [12, 13]. To address these limitations, it is crucial to develop RE solutions that are grounded in industrial practice and incorporate comprehensible explanations along with output to support effective human-in-the-loop validation.

Building on this, the primary goal of the thesis is to enhance decision support for practitioners by facilitating the management and analysis of requirements, thereby reducing their workload in developing complex industrial systems. In this regard, we collaborated closely with Alstom Rail AB[1] (Alstom), a global leader in railway manufacturing, to investigate AI-based techniques for enhancing their RE practices. In particular, we address four primary industrial challenges: (i) identifying requirements from tender documents, (ii) detecting requirement ambiguities, (iii) allocating requirements across subsystem teams, and (iv) addressing requirement-related queries

[1] https://www.alstom.com/alstom-sweden

during the release management process. These tasks are predominantly carried out manually, making them time-consuming and prone to inconsistencies. By proposing AI-driven solutions, the work enhances the efficiency of RE tasks, thereby supporting the development of complex industrial systems in the studied setting.

Thesis Outline. This thesis consists of two parts. Part I gives an overview of the thesis and is organized as follows. Chapter 2 discusses the preliminaries and related work to the thesis. Chapter 3 presents the research overview, including two subsections: (i) research context and goals, and (ii) research process. Chapter 4 discusses the research results, mainly the thesis contributions. Finally, Chapter 5 provides the conclusion of the thesis and future work. Part II contains the complete set of research papers that form the foundation of the thesis. The papers have been reformatted to ensure consistency with the overall thesis structure.

Chapter 2

Preliminaries and Related Work

This chapter consists of two main sections: Preliminaries and Related Work. In preliminaries, we discuss the central topics of this thesis, including requirements engineering, AI for requirements engineering, and NLP pipeline and techniques. The related work section reviews the state-of-the-art and provides a reflection on the existing body of knowledge.

2.1 Preliminaries

2.1.1 Requirements Engineering

Requirements Engineering (RE) is a widely recognized discipline marked by well-established concepts within the system and software engineering domains. RE involves elicitation, analysis, validation, and management of requirements for complex systems across all stages of development. Specifically, the role of RE in safety-critical environments, such as railways, is crucial as it directly impacts the system safety and regulatory compliance [14].

A notable early contribution in the RE field comes from Alford et al. [15], who introduced the Software Requirements Engineering Methodology (SREM), designed for developing software-intensive defense systems. However, the establishment of a formal international standard came at a later stage. The comprehensive ISO life-cycle standard (ISO/IEC/IEEE 29148:2011 [16]) defines the RE as *"an interdisciplinary function that mediates between the domains of the acquirer and supplier to establish and maintain the requirements to be met by the system, software or service [...]"*. This highlights the role of RE methods in bridging the gap between problem and solution space, aligning with user expectations and regulatory constraints.

At the center of RE are the requirements themselves, articulated as a threefold concept by the International Requirements Engineering Board (IREB) as *"1. A need perceived by a stakeholder. 2. A capability or property that a system shall have. 3. A documented representation of a need, capability, or property."* [17]. RE practices aim to create a set of requirements that are agreed upon by stakeholders, validated against the needs, and usable as a reference for system implementation and verification.

Despite the existence of standards and formal methods, industrial RE remains a complex and context-specific activity. Practitioners often face challenges, including but not limited to ambiguities in natural language specifications, difficulties in maintaining traceability across evolving artifacts, and classifying requirements for management [11]. These challenges often contribute to project delays and costly rework in large-scale, regulated environments, such as the railway sector, where requirements usually span multiple subsystems and interact with diverse engineering domains.

In response, there has been a growing interest in applying AI techniques, particularly NLP methods based on language models, to augment various RE tasks. This need stems from the continuous increase in system complexity and technological advancements. Additionally, the scope and scale of developing systems have expanded significantly, necessitating enhanced support in the traditional RE process to align with stakeholder expectations [1].

2.1.2 AI for Requirements Engineering

Over the last decade, AI-based technologies have evolved rapidly, making significant progress in supporting a range of software engineering tasks [18, 19]. Within the realm of RE, the application of AI, specifically advanced NLP algorithms based on language models can support various stages of the RE process. This is because language models can process and interpret unstructured textual data effectively [20], which is often the case for most requirements documents.

Recent advances in NLP have enabled automation in various RE tasks, including defect detection [21, 22], classification [23, 7], traceability [24, 25], and reuse [26]. This is crucial and relevant in the industrial context, where RE practices are predominantly manual and tedious. Since natural language remains the de facto standard for specifying requirements, NLP techniques that can interpret and reason over such text have become increasingly valuable in addressing industrial RE challenges [10]. In this regard, a recent systematic mapping study by Zhao et al. [12] has categorized the applications of NLP in RE into six main tasks: detection, extraction, classification, modelling, tracing & relating, and search & retrieval. The study surveyed more than 400 primary studies and identified over 230 NLP technologies, mapping them according to RE phases and task categories to highlight the growing role of AI methods in aiding system development. Additionally, it demonstrates that a range of NLP pipelines with varying configurations can be utilized to construct effective RE solutions.

2.1.3 NLP Pipeline and Techniques

Broadly, the development of NLP-based solutions for various RE tasks can be categorized into four stages: Preprocessing, Representation, Modeling and Evaluation. Below, we discuss each stage briefly.

Preprocessing

The automation of RE tasks through NLP-based solutions requires applying a language model that represents text as vectors in a continuous space [20]. These vector representations are employed as features to perform downstream tasks such as finding similarities [27] and classifying requirements into different categories [28]. Since raw text, like natural language requirements, is typically unstructured and contains noise, a preprocessing stage is a crucial first step in preparing the data for feature representation in later stages. In general, preprocessing techniques include, but are not limited to, removing stop words (e.g., ‘the’, ‘is’), lemmatization to reduce words to their base forms, tokenization, and part-of-speech tagging [29, 30]. However, the importance and application of preprocessing techniques depend on the downstream task and the unit of analysis, which can be words, sentences, or paragraphs. For instance, traditional machine learning models often require removing stop words to reduce noise and dimensionality. In contrast, deep learning approaches may tend to consider such words, as they contribute to the contextual significance and model performance.

Representation

To enable different NLP-based language models to understand and interpret natural language requirements, a representation layer is required to transform input text into a structured numerical form. This representation incorporates syntactic, semantic, and contextual information derived from the learned language model, which is crucial for downstream tasks. For instance, categorical representations such as Bag-of-Words and weighted techniques like TF-IDF [31] capture word frequency and document-level importance, but are unable to capture contextual or semantic relationships [32]. On the other hand, distributed representations, such as GloVe [33] and FastText [34] techniques, encode words in dense, low-dimensional vector spaces that preserve semantic and syntactic structures. However, these embeddings are static, where each word has a single vector regardless of context, while embedding techniques like Sentence-BERT [35] and Universal Sentence Encoder [36] generate context-aware sentence-level embeddings. These embeddings can be directly used for tasks like identifying semantically similar requirements for reuse [18], and they can also serve as input features for supervised models for requirements classification [37].

Modeling

In the modeling stage, the representations of natural language requirements are transformed into structured inputs that enable models to make probabilistic inferences and generate outputs. These models range from traditional machine learning (ML) to advanced transformer-based architectures, which are tuned to perform domain-specific tasks. In the context of RE, such models are trained on relevant features to support tasks like requirements classification and retrieval [12]. The main objective of this stage is to learn the textual patterns and semantic structures to enable automated analysis, however, the selection of the modeling technique depends on the downstream task. Below, we briefly discuss the evolution of modeling techniques from traditional to current state-of-the-art for RE tasks within the scope of this thesis.

Traditional Machine Learning Models. Classical ML models for supporting RE tasks typically rely on syntactic features for text representation. Standard techniques include Bag-of-Words, TF-IDF weighting, n-gram extraction, and syntactic patterns like part-of-speech tags to encode requirement statements as fixed-length feature vectors [32]. These features are used to train ML classifiers, such as decision trees, random forests, logistic regression, naive bayes, and support vector machines, for tasks like requirement classification [38]. While these models are effective to some extent, they have shown fundamental limitations in capturing semantic similarity, linguistic variations, and contextual dependencies. As a result, they struggle to capture the inherent complexities in domain-specific natural language requirements.

Deep Learning Models. To address the limitations of contextual dependencies and semantic similarity in classical ML models, deep learning approaches based on recurrent neural networks have been widely adopted for RE tasks. Notably, long short-term memory (LSTM) networks [39] have been commonly employed to model the sequential nature of requirement statements by effectively capturing token dependencies. These models are typically trained on dense pre-trained semantic embeddings such as GloVe [33] or FastText [34]. FastText has been quite useful for domain-specific tasks because of its ability to handle subword information, which addresses the issue of out-of-vocabulary terms. Despite achieving improved semantic understanding compared to classical approaches, these deep learning models still require large, labeled datasets to perform well.

Transformer-based language models. With the advent of transformer-based architectures, there has been a fundamental shift in NLP applications by replacing sequence learning with a self-attention mechanism [40]. The self-attention mechanism has enabled language models to capture long-range dependencies and global context more effectively. The transformer architecture is the foundation of pre-trained LLMs, such as Bidirectional Encoder Representations from Transformers (BERT) [41] and its variants like RoBERTa [42], which have been widely adopted for a range of RE tasks like classification, ambiguity detection, and traceability link recovery [43, 44]. In contrast to LSTM-based models, which process input sequentially, transformer models compute contextualized representations for each token by attending to all others in parallel. This makes them particularly well-suited for analyzing long and syntactically complex requirement texts.

Recent advancements in transformer architectures have focused on autoregressive language modeling, where the model generates the text by predicting the next token given the preceding context. Large autoregressive models such as the Generative Pretrained Transformers (GPT), notably GPT-4 [45], have demonstrated state-of-the-art performance in a range of language generation tasks such as dialogue systems. In parallel, open-source generative LLMs like Llama [46] and Qwen [47] series adopt similar autoregressive designs and can be utilized for downstream tasks like requirements classification and analysis [48]. In addition, the instruction-tuned variants of these models show better alignment with user intent and often require a few-shot examples to perform the downstream task effectively [49]. The ability of recent LLMs to perform well with few example demonstrations has addressed the limitations of earlier models that relied mainly on huge task-specific annotated corpora. This is particularly relevant in the RE domain, where labeled data is often scarce.

Evaluation

In the final stage of the NLP pipeline, it is critical to assess the effectiveness, reliability and generalization of models in real-world applications. The choice of evaluation criteria is solely task-dependent and must align with the objectives of the downstream applications. In industrial, domain-specific settings, quantitative metrics are often coupled with qualitative assessments to provide a comprehensive evaluation. For instance, in the requirements classification task, standard metrics such as precision, recall, and F1 scores are combined with human assessments to gain practical insights into model performance [50, 51, 52]. In addition, qualitative assessments, such as focus groups or feedback on results, can be systematically analyzed to identify limitations and areas for improvement in the developed pipelines [53, 54]. Similarly, for tasks like information retrieval in chatbot-based systems, evaluation must consider semantic correctness, contextual relevance, and practical usability. This is because traditional metrics, such as BLEU and ROGUE, which focus on lexical overlap, are inadequate for capturing the semantic accuracy of responses [55]. Therefore, such metrics can be complemented with expert human evaluations to assess whether the outputs are factually correct and usable in industrial settings [56].

2.2 Related Work

The research conducted in this thesis can be categorized into two main themes: requirements classification and requirements information retrieval. The first theme includes various RE tasks formulated as classification problems. These tasks include requirements identification in tender documents, requirements ambiguity detection with explanations, and allocation of requirements to multiple teams. The second theme focuses on retrieval-based methods to answer requirements-related queries by extracting relevant information from technical documentation. Below, we discuss the prior work related to each theme and our reflection on the state-of-the-art.

2.2.1 Requirements Classification

In the context of RE, classification is one of the most common and essential activities that involves organizing and interpreting unstructured natural language requirements for downstream analysis and use [12]. Due to the inherent variability in natural language, requirements classification techniques are designed to consider linguistic variations and domain-specific terminologies and concepts. In this regard, the literature consists of multiple techniques for classification, ranging from traditional rule-based methods to advanced language models. In the following, we discuss the related work of three RE cases addressed in this thesis: requirements identification, ambiguity detection, and allocation.

Requirements Identification. The identification of requirements is often treated as a classification problem in the literature [57]. Within this, NLP pipelines based on traditional ML algorithms are employed to separate requirements from other supporting information. In this regard, a notable contribution comes from the work by Abualhaija et al. [57] to demarcate requirements from other information in large requirement specification documents. The authors considered each sentence in the documents as a unit for classification and trained supervised ML models on 33 documents, achieving a recall score of 95% in distinguishing requirements from auxiliary information. Additionally, Sainani et al. [58] proposed a two-stage methodology that initially employs Bi-directional Long Short-Term Memory networks for extracting requirements from contractual documents, followed by the utilization of BERT LLM for their subsequent allocation into sub-classes. Similarly, Winkler et al. [59] utilized a Convolutional Neural Networks to identify requirements from other supporting material within IBM DOORS repositories. Falkner et al. [60] adopt a Naive Bayes classifier trained on word distributions to identify requirements within Request for Proposal (RFP) documents from the railway domain. On the other hand, Jindal et al. [61] proposed an approach for extracting and classifying security requirements, using TF-IDF vectors and a Decision Tree algorithm. Varenov et al. [62] introduced a sentence-level classifier based on a fine-tuned DistilBERT model to categorize security requirements into predefined groups.

Requirements Ambiguity Detection. Ambiguity in requirements also known as smells [63] or defects [22] has been widely studied in literature. A seminal theoretical work in this area is the handbook by Berry et al. [64], which categorizes ambiguities mainly into four different types: lexical, syntactic, semantic, and pragmatic. Similarly, Gervasi et al. [65] provided a unified framework for ambiguity in requirements specifications and interviews. On the other hand, several methods and tools have been developed to detect ambiguities in requirements. In this regard, Bucchiarone et al. [66] and Tjong et al. [67] proposed QuARS and SREE tools, respectively, which automatically identify ambiguous keywords or key-phrases in the requirements text. Ferrari et al. [68, 69] utilized knowledge graphs and search algorithms to detect pragmatic ambiguities arising from contextual interpretation. In another work, Veizaga et al. [70] proposed the Paska tool based on rule-based NLP techniques combined with a controlled natural language to detect requirements smells within the financial domain. Yang et al. [71] applied a set of heuristics combined with a k-nearest neighbors algorithm to find anaphoric ambiguities associated with pronouns such as *it*, *they*, etc. Likewise, Beqiri et al. [51] proposed a hybrid approach that combines NLP techniques with traditional ML algorithms to detect ambiguity in the railway domain, supporting the identification of potentially ambiguous terms. More recently, Yildirim et al. [72]

employed generative LLMs for addressing ambiguity in RE, demonstrating that these models outperform traditional classification approaches.

Requirements Allocation. The automated requirements classification has become a central task in the RE literature, second only to defect detection, as highlighted by Zhao et al [12] in a systematic mapping study. In the context of this thesis, we formulated the task of requirements allocation as a multi-class classification problem. Within this group, the initial contribution comes from Cleland-Huang et al. [73, 74], who proposed a keyword-based classification method to identify a set of non-functional requirements (NFRs) and released the PROMISE dataset [75], which later became a standard benchmark. Kurtanović and Maalej [8] applied SVM to this dataset, for requirements classification, achieving performance above 90% in terms of precision and recall. Dalpiaz et al. [7] expanded this work using manually labeled datasets and linguistic features, improving the explainability of the classification algorithms. To address the limitations of manual feature engineering, researchers increasingly adopted deep learning based techniques. Navarro et al. [76] employed Convolutional Neural Networks with pre-trained Word2Vec embeddings, while Aldhafer et al. [77] employed Bidirectional Gated Recurrent Neural Networks to perform requirements classification. Recently, Hey et al. [78] performed transfer learning with BERT LLM and achieved state-of-the-art results compared to traditional ML on requirements classification using the PROMISE benchmark dataset. On the other hand, limited literature is available that explicitly addresses the problem of classifying requirements into sub-classes to facilitate allocation for architecture decomposition, which is one of the key tasks studied in this thesis. Within the group of allocation, the work has largely been explored in issue tracking, where Aktas et al. [79] developed an SVM-based classifier to assign issues to teams in the financial sector, achieving an F1 score of 80%. Similarly, Jonsson et al. [80] employed an ensemble learning approach on datasets from two companies, achieving an accuracy of 89% for issue allocation.

2.2.2 Requirements Information Retrieval

Information Retrieval (IR) techniques are designed to identify and extract relevant information from large corpora based on user queries [30]. Within the

RE domain, IR has been a foundational approach to support practitioners in locating critical information across various resources and, in turn, facilitating tasks like traceability and reuse [81]. The role of IR has become more critical with the advent of LLMs, which, despite their advanced capabilities in performing generalized tasks, often provide factually incorrect information or hallucinate in specialized domains [82]. To address this, IR-based techniques are increasingly employed with LLMs, commonly known as Retrieval Augmented Generation (RAG), for retrieving domain-specific relevant information and providing it as contextual input to improve the accuracy and reliability of the generated responses. In the following, we discuss the related work on the RE case addressed in this thesis, which is a domain-specific chatbot that answers requirement-related queries from technical documentation.

Recent studies have explored the use of RAG-based frameworks in various software engineering tasks, such as test case generation and code comprehension [83]. Ali et al. [84] introduced a RAG-based chatbot leveraging Llama 3 LLM to improve traceability between natural language requirements and source code. Similarly, Ezzini et al. [85] proposed a tool, QAssist, a question-answering assistant for improving requirements quality through automated analysis and feedback. QAssist is based on a domain-specific corpus and an external knowledge base, along with BERT-based LLMs to highlight relevant passages given user queries. In another work, Arora et al. [86] proposed RAGTAG, which generates test cases using natural language requirements. The approach considered zero-shot and few-shot prompting techniques using the GPT models. Chaudhary et al. [87] developed a RAG-based chatbot to assist with queries related to CI/CD documentation. For context retrieval, the authors compared traditional IR techniques, i.e., TF-IDF and BM25, which were coupled with a Llama 2 LLM to generate domain-specific responses.

2.2.3 Reflection on the Related Work

In literature, AI-based techniques have shown strong capabilities across several crucial RE tasks, including requirements classification and retrieval [12]. A wide range of approaches has been explored, including traditional ML, deep learning, few-shot learning, and retrieval-augmented LLMs. However, despite the advancements, the practical relevance of these approaches in industrial contexts remains limited. Fernández et al. [11] highlights this gap, indicating that only 16% of the companies are using automated techniques in their RE process. The underutilization of existing proposed solutions in industrial settings is primarily due to (i) the limited evaluation using public datasets annotated by students or researchers (e.g., PROMISE [75]), which do not capture the complex needs of the specific industrial domain, and (ii) lack of confidence in developed solutions, particularly when techniques do not provide comprehensible rationales or additional augmentations to facilitate human-in-the-loop validation. This is critical in complex industries like the railway sector, where technological solutions are expected to augment and support practitioners' activities rather than fully automate or replace their decision-making.

This thesis addresses these gaps by empirically evaluating a wide range of AI-based techniques across multiple RE tasks on industrial datasets, developing solutions that are both practically relevant and aligned with the needs of requirement-centered domains, such as the railway industry, to support practitioners in making well-informed and efficient decisions.

Chapter 3

Research Overview

In this chapter, we first present the research context and goals of the thesis, and then outline the research process designed to achieve these goals.

3.1 Research Context and Goals

In the railway industry, the growing complexity of large-scale software-intensive systems, primarily driven by technological advancements and evolving stakeholders' expectations, emphasizes the need for an efficient RE process. During the development of such safety-critical systems, the requirements evolve through various levels of abstraction and atomicity. Initially appearing as high-level customer specifications and broad concepts, they evolve into detailed, granular-level requirements during the RE process. This evolution of requirements is particularly challenging in the railway industry, where systems are decomposed into multiple interdependent subsystems, each with distinct development specifications and regulatory constraints. In addition, the prevalent use of manual practices increases the effort required to identify, allocate, and trace requirements across different system levels, making the process tedious, prone to errors, and frequently leading to delays in time-to-market. To minimize project risks and ensure timely deliveries to customers, companies are seeking solutions that can enhance efficiency and reduce reliance on manual efforts. As a result, there is an evident need

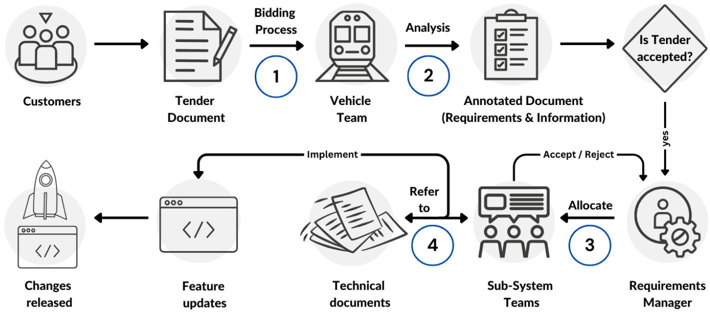

Figure 3.1: Overview of requirements flow in the studied context

for advanced automated tools to support practitioners' decision-making in performing crucial RE tasks and, in turn, managing the growing complexity of systems more efficiently. To this end, we define the overall research goal of the thesis as follows:

> **Research Goal** (RG): To enhance decision support in requirements management for the development of complex industrial systems.

To realize the overall RG of the thesis, we worked in close collaboration with our industrial partner from the railway industry, Alstom. Figure 3.1 shows the requirements flow and its evolution in the studied setting, providing context for the practical challenges examined through multiple case studies.[1] The RE process at Alstom commences with analyzing the tender documents received from customers. This process, traditionally manual and time-intensive, is crucial for enabling the project acquisition and impacts the downstream RE activities. During the bidding phase, the vehicle team receives the tender document, often consisting of 600 to 1800 individual segments of natural language text containing high-level technical specifications (requirements), supporting information, and contractual obligations. Given the substantial volume of tender documents, the vehicle team dedicates significant time and effort to analyzing and distinguishing requirements from other supporting information (Step 1). Further, the analysis is prone to ambiguities due to (i) the diverse backgrounds of stakeholders and (ii) the inherent variability of natural language used to

[1] The four steps shown in Figure 3.1 correspond to the focus areas of the industrial case studies and form the basis for the main contributions of the thesis.

specify requirements. These ambiguities can lead to misinterpretations and increased rework costs (Step 2), which makes early detection and clarification during the RE process crucial to avoid scope creep and ensure alignment in large-scale projects.

Later, once the tender bid is accepted and the requirements are agreed upon, they are assigned to multiple subsystem teams for implementation and testing. This shows the inherent complexity of railway platforms, which are composed of multiple interdependent subsystems. However, this architecture provides modularity and better control over the process, as the components can be independently implemented, verified, and incrementally integrated. Typically, a requirements manager manually allocates requirements across nearly 15 teams for development and rigorous testing, making the process time-consuming (Step 3). Additionally, if teams are assigned requirements unrelated to their own, they are rejected, resulting in extended feedback cycles. During requirements development—specifically when introducing new versions or updates—engineers not only support end-users by integrating new updates with legacy components and resolving compatibility issues, but also ensure that updates remain consistent with system-level requirements. For this purpose, engineers often refer to technical documents such as release notes, architectural and interface control documents, to ensure compliance and that each new requirement update aligns with the existing system-level requirements. However, manually retrieving relevant information is tedious and error-prone, as the required information may be scattered within huge documents, resulting in inefficiencies during the release management (Step 4). Therefore, it is essential to provide effective solutions that can address these challenges, supporting practitioners in making decisions and, in turn, efficiently managing requirements for the development of complex systems.

To this end, we break down the RG into two research subgoals, each focusing on practical problems related to the requirements management process in the studied context. Specifically, the research subgoal 1 (RSG_1) aims to provide decision support solutions that focus on addressing the challenge of distinguishing requirements from other information in tender documents, detecting and resolving requirements ambiguities (Steps 1 and 2 in Figure 3.1) to aid the practitioners in project acquisition. RSG_1 is defined as follows:

> RSG_1: To enhance the efficiency of industrial practitioners in the **identification and disambiguation of requirements** from tender documents during project acquisition.

Research subgoal 2 (RSG_2) aims to improve decision support for practitioners by assisting in allocating requirements across multiple teams and facilitating their development (Steps 3 and 4 in Figure 3.1) and is defined as follows:

> RSG_2: To enhance the efficiency of industrial practitioners in **allocating requirements across teams and their development** during project release.

3.1.1 Research Process

To produce findings that are both valuable and relevant to academia and industry, it is necessary to employ a process designed to support industry-academia collaboration. Accordingly, in this thesis, we employ the constructive research methodology [88] as our primary research approach, which supports this collaborative context. The constructive research methodology aims to solve real-world problems by combining theoretical insights with practical relevance and provides a structured process that enables close collaboration between researchers and industry by identifying relevant challenges, co-developing solutions through iterative cycles, and supporting the validation and transfer of research outputs in industrial settings. Figure 3.2 has been adapted to show this process in the context of this thesis, with a concise summary of each step outlined below.

1. Identification of Industrial Challenges. The first step is to identify challenges of practical relevance. For this purpose, we investigate the existing RE practices by performing document analysis and assessing the needs of our industrial partner (Alstom) within the scope of AIDOaRt[2] and SmartDelta[3]

[2] AIDOaRt is a 3-year European project and was started on 2021-04-01. The project focuses on AI-enhanced automation to support the development lifecycle of cyber-physical systems. (https://sites.mdu.se/aidoart)

[3] SmartDelta is a 3-year ITEA project and was started on 2021-12-01. The project focuses on quality assurance and optimization in the development of incremental industrial software systems. (https://smartdelta.org)

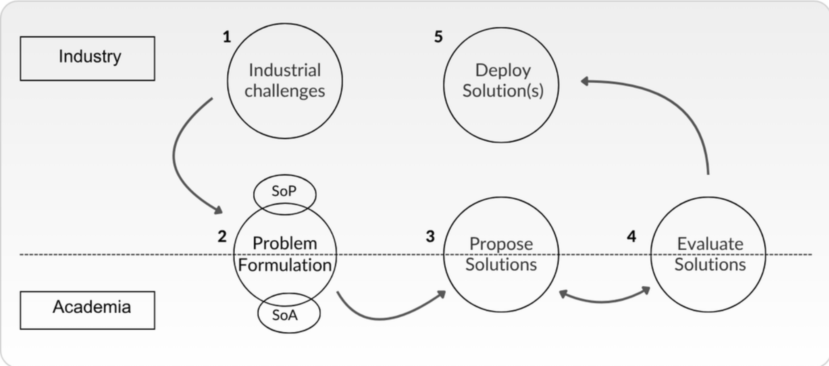

Figure 3.2: Research process adapted to our context

projects. As a result, we identify a set of critical research challenges in the RE process, which in turn, led to the formulation of the overall research goal and subgoals.

2. Problem Formulation. We formulated research problems through close collaboration with Alstom, beginning with a series of brainstorming sessions that involved industrial champions and practitioners. We further engaged in several AIDOaRt hackathons[4] [5] with Alstom, where challenges were collaboratively refined into concrete research problems. An initial formulation of the problems and their industrial relevance also resulted in a doctoral symposium paper [89].

3. Propose Solutions. For the realization of RSG_1 and RSG_2, we proposed and evaluated multiple AI-based solutions, resulting in working prototype tools. We performed empirical evaluations for each specific use case, selecting and comparing techniques—ranging from traditional ML and deep learning to state-of-the-art LLMs—based on task complexity and dataset characteristics to determine the most effective solution. Specifically, to support practitioners during the project acquisition phase (RSG_1), we proposed two solutions to identify and disambiguate requirements. The first, *REQ-I*, utilizes

[4] https://sites.mdu.se/aidoart/events/first-hackathon
[5] https://sites.mdu.se/aidoart/events/third-hackathon

a BERT-based LLM to identify requirements from tender documents. The second, *RADICLE*, relies on a few-shot in-context learning paradigm with LLMs to identify ambiguous requirements and provide additional explanations to support clarifications. Similarly, to assist practitioners in allocating requirements across teams and managing new updates during project releases (RSG_2), we also developed two solutions. *REQA* combines a BERT-based LLM with traditional ML techniques—including k-nearest neighbors (kNN) and k-means clustering [90]—to allocate requirements to subsystem teams and generate additional augmentations to support informed decision-making. *ReqRAG* adopts a RAG-based technique using LLMs to provide contextually relevant answers to queries related to requirement changes, and in turn, supports their development and release phases.

4. Evaluate Solutions. We performed a two-stage evaluation of the proposed solutions. First, we developed and quantitatively evaluated all the solutions in laboratory settings at RISE Research Institutes of Sweden, using locally hosted servers to measure performance. Second, the *REQ-I* and *REQA* tools were evaluated in collaboration with Alstom to assess their effectiveness in real-world settings. The other two solutions, *RADICLE* and *ReqRAG*, underwent expert review, where industry practitioners reviewed and rated the quality of their outputs and provided feedback for future improvements.

5. Deploy Solution(s). We deployed the *REQ-I* and *REQA* tools at Alstom, as Docker-containerized applications, each with a simplified user interface. The requirements engineers evaluated the tools and during the final use case assessment of the AIDOaRt project, Alstom reported that both tools can reduce manual efforts by approximately 80% on average compared to the manual baseline[6]. For *RADICLE* and *ReqRAG* solutions, we plan to deploy them in the future.

[6]For further details, see: D5.9. Use Cases Evaluation Report - 2: Use case scenario BT_UCS1, pp. 95-103 (`https://sites.mdu.se/aidoart/results/deliverables`)

Chapter 4

Research Results

In this chapter, we discuss the results achieved in this thesis. First, we present the thesis contributions and then outline papers corresponding to the contributions.

4.1 Thesis Contributions

This thesis addresses the research goals discussed in 3.1 through four main research contributions (RC_x), each resulting in the publication referred to as Paper, labelled A through D. Figure 4.1 outlines the mapping between the research goals and contributions and the resulting Papers. Below, we discuss the four (RC_x) in detail.

- **RC_1:** *Identification of requirements within tender documents.*
- **RC_2:** *Detection and explanation of requirements ambiguities.*
- **RC_3:** *Context-aware requirements allocation to multiple teams.*
- **RC_4:** *Retrieval of requirements knowledge during software releases.*

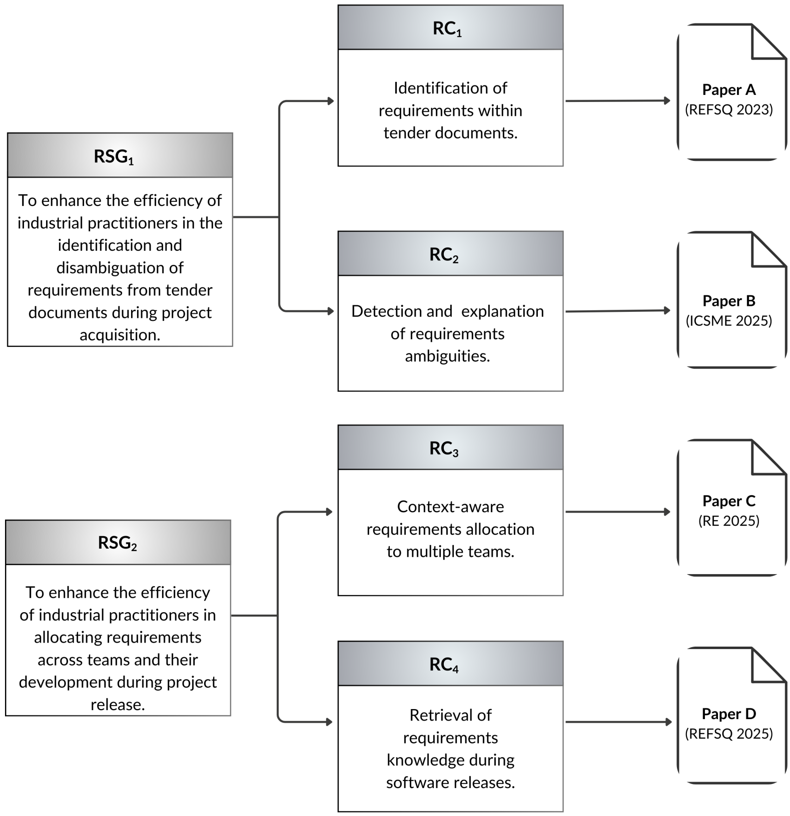

Figure 4.1: Mapping of research goals to contributions to papers

4.1.1 RC_1: Identification of requirements within tender documents

The analysis of tender documents is a critical step, as it enables the risk estimation and time planning for project development. In practice, however, identifying requirements within the tender documents is both time-intensive and error-prone, which can cause cascading effects on the downstream development process. To address the challenge of distinguishing requirements from auxiliary information in large railway tender documents, this research contribution (RC_1), proposes an approach based on LLMs for requirements identification, resulting in the publication of Paper A [37].

We formulate the task of requirements identification as a binary text classification problem. In this regard, we conduct a comprehensive empirical evaluation of 20+ classification pipelines, ranging from traditional machine learning and deep learning approaches to transformer-based models on both industrial and public datasets. We evaluate the annotated dataset provided by our industrial partner, Alstom, extracted from five tender documents used in real-world railway projects. Additionally, we utilize a publicly available dronology dataset to support reproducibility and evaluate the generalizability of the considered pipelines beyond the studied context. The results from the evaluation show that transformer-based models, particularly the BERT-based uncased model, achieved the highest average F1 score of 82% on the industrial dataset and 87% on the public dataset. These findings informed the development of the *requirement identifier (REQ-I)* tool, which leverages the best-performing pipeline to classify requirements in tender documents within the studied settings. Furthermore, internal evaluation of *REQ-I* tool by Alstom engineers indicates that it can reduce manual effort by 80% on average, highlighting the practical utility of the proposed solution in supporting practitioners during the project acquisition process.

4.1.2 RC_2: Detection and explanation of requirements ambiguities

Natural language is the de facto standard for specifying requirements. However, it is inherently prone to ambiguities, often leading to misunderstandings among stakeholders and project delays. Therefore, detecting and explaining potential requirements ambiguities early in the project acquisition process can enable practitioners to resolve misunderstandings before they propagate to downstream development phases. To address this, the research contribution (RC_2), presented in Paper B [50], proposes an approach that leverages an in-context learning paradigm with LLMs to automatically detect ambiguous requirements and generate human-like explanations, thereby supporting early clarification.

We frame the problem as an explanation-augmented binary classification task, where promote templates are constructed using task-specific demonstrations (labeled examples of ambiguous and unambiguous requirements) combined with a query (a new requirement to be classified and explained). This formulation leverages the few-shot generalization capabilities of LLMs to perform both classification and generate human-like explanations, without requiring model fine-tuning. We conduct an empirical evaluation of our in-context learning approach on two industrial datasets from Alstom and one from Westermo Network Technologies[1] using various open-source LLMs. In addition, the work examines the effect of demonstration quantity (zero-shot, one-shot, five-shot and ten-shot) on model performance and includes a human evaluation with industry practitioners to gain practical insights. The results show that the in-context few-shot learning significantly improves ambiguity classification performance by an average of 20.2% compared to zero-shot demonstrations. Furthermore, in an expert evaluation conducted with practitioners from the two companies, the generated explanations by LLMs were found to be natural, adequate, useful and relevant in 77% of the instances on average.

4.1.3 RC_3: Context-aware requirements allocation to multiple teams

Complex industrial systems, such as those in the railway domain, are composed of multiple interacting subsystems to ensure architectural modularity and structured development. Allocating requirements to their specific subsystem teams for development and testing, however, is a challenging, error-prone and time-consuming task. This is primarily because large-scale railway platforms comprise over 15+ subsystems, requiring a significant volume of requirements that must be allocated to their respective teams. In practice, misallocations are common and often lead to rejections, resulting in longer feedback loops and delays in the development process. To address this, the third research contribution (RC_3) presents a context-aware approach, the *requirement allocator (REQA)*, consisting of two modules: *Assigner* and *Augmenter*. *REQA* classifies input requirements into the appropriate team—a multi-class classification task—using a language model (*Assigner*), and complements these predictions with additional information—which are similar requirements—derived from lexical similarity measures (*Augmenter*). In particular, the *Augmenter* represents each requirement as a vector, uses k-means to locate the nearest cluster, and retrieves the most similar requirements using k-Nearest Neigh-

[1] `https://www.westermo.com/se`

bors (kNN). The *Augmenter* module enhances the capability of practitioners in making well-informed decisions during the allocation of requirements.

The RC_3 resulted in the publication of Paper C [52], consisting of an empirical evaluation of the requirements allocation task. In particular, we compared multiple techniques, including traditional machine learning approaches and state-of-the-art transformer-based models, to identify the best-performing approach for *REQA*, in collaboration with our industrial partner, Alstom. Results show that transformer-based models, particularly BERT variants, performed the best in terms of the F1 scores. The *Assigner* module achieved an average F1 score of 76% for the five most frequent teams, while the *Augmenter* provided correct and useful indications in 76% of the cases. We deployed *REQA* at Alstom, and an internal evaluation indicated that *REQA* can potentially reduce manual efforts by 80% on average, showing its practical usefulness in enhancing the efficiency of practitioners in performing the requirements allocation task.

4.1.4 RC_4: Retrieval of requirements knowledge during software releases

Release management facilitates the traceability of software changes across versions, which is crucial to ensure that each iteration of the system aligns with specified requirements and compliance standards. In practice, however, such critical information is scattered across multiple technical documents, such as release notes and interface control documents, which engineers must manually cross-reference to perform verification and compatibility analysis. This retrieval of information is often manual and time-intensive because of the large scale of the documents. To address this challenge, the fourth and final research contribution, RC_4, of this thesis introduces a domain-specific solution, *ReqRAG*, developed in collaboration with Alstom, leading to the publication of Paper D [5].

ReqRAG employs a retrieval augmented generation technique, combined with LLM, to provide accurate and context-aware responses based on Alstom's internal documentation. In particular, *ReqRAG* constitutes two phases: *Retrieval* and *Generation*. In the *Retrieval* phase, we transform technical documents into a searchable knowledge base. In this regard, we employ (i) an OCR model to extract textual data from complex PDF documents, (ii) preprocess the data to remove noise such as headers, and (iii) segment the text into semantic chunks for retrieval. These chunks are encoded into vector embeddings and indexed in a vector database. Note that the retrieval steps are performed once during the setup and are not repeated at each query time. In the *Generation* phase, the relevant context chunks are retrieved for a given query and incorporated into a prompt template that includes system-level instructions—defining the guidelines to ensure that the responses are well-aligned with the domain context—and then we invoke the LLM to generate a context-aware domain-specific response. To select the best-performing configuration for *ReqRAG* phases, we evaluated multiple OCR models, embedding techniques and open-source LLMs. The *Retrieval* phase achieved a 90% context recall score on the Alstom dataset. Additionally, we conducted a human evaluation, where industry experts found, on average 70% of the generated responses to be useful, adequate, and relevant in addressing the queries.

4.2 Paper Contributions

This section provides a brief overview of the papers included in the thesis. Figure 4.1 shows the mapping between the research contributions and the resulting papers.

4.2.1 Individual Contributions

In Papers A and C, I was a co-driving researcher alongside the second author, with equal contributions from conceptualization to implementation. In Papers B and D, I was the primary driving researcher. Across all papers, the remaining co-authors contributed through design discussions, reviews, and iterative improvements.

4.2.2 Included Papers

Paper A: Requirement or Not, That is the Question: A Case from the Railway Industry
Authors: *Sarmad Bashir*, Muhammad Abbas, Mehrdad Saadatmand, Eduard Paul Enoiu, Markus Bohlin, Pernilla Lindberg.

Status: Published in the 29th International Working Conference on Requirement Engineering: Foundation for Software Quality (REFSQ 2023).

Abstract: **[Context and Motivation]** Requirements in tender documents are often mixed with other supporting information. Identifying requirements in large tender documents could aid the bidding process and help estimate the risk associated with the project. **[Question/problem]** Manual identification of requirements in large documents is a resource-intensive activity that is prone to human error and limits scalability. This study compares various state-of-the-art approaches for requirements identification in an industrial context. For generalizability, we also present an evaluation on a real-world public dataset. **[Principal ideas/results]** We formulate the requirement identification problem as a binary text classification problem. Various state-of-the-art classifiers based on traditional machine learning, deep learning, and few-shot learning are evaluated for requirements identification based on accuracy, precision, recall, and F1 score. Results from the evaluation show that the transformer-based BERT classifier performs the best, with an average F1 score of 0.82 and 0.87 on industrial and public datasets, respectively. Our results also confirm that few-shot classifiers can achieve comparable results with an average F1 score of 0.76 on significantly lower samples, i.e., only 20% of the data. **[Contribution]** There is little empirical evidence on the use of large language models and few-shots classifiers for requirements identification. This paper fills this gap by presenting an industrial empirical evaluation of the state-of-the-art approaches for requirements identification in large tender documents. We also provide a running tool and a replication package for further experimentation to support future research in this area.

Paper B: Requirements Ambiguity Detection and Explanation with LLMs: An Industrial Study

Authors: *Sarmad Bashir*, Alessio Ferrari, Abbas Khan, Per Erik Strandberg, Zulqarnain Haider, Mehrdad Saadatmand, Markus Bohlin.

Status: Published in the 41st International Conference on Software Maintenance and Evolution (ICSME 2025).

Abstract: Developing large-scale industrial systems requires high-quality requirements to avoid costly rework and project delays. However, linguistic ambiguities in natural language (NL) requirements have been a long-standing challenge, often introducing misinterpretations and inconsistencies that propagate throughout the development lifecycle. Such ambiguous NL requirements necessitate early detection and well-reasoned explanations to clarify and prevent further misunderstandings among stakeholders. While solutions have been developed to detect ambiguities in NL requirements, the advent of generative large language models (LLMs) offers new avenues for explanation-augmented requirements ambiguity detection. This paper empirically investigates LLMs for ambiguity detection and explanation in real-world industrial requirements by adopting an in-context learning paradigm. Our results from three industrial datasets show that LLMs achieve a 20.2% average performance increase in classifying ambiguous requirements when prompted with ten relevant in-context demonstrations (10-shot), compared to no demonstrations (0-shot). Additionally, we conducted human evaluations of the LLM-generated outputs with eight industry experts along four dimensions—naturalness, adequacy, usefulness and relevance—to gain practical insights. The results show an average rating of 3.84 out of 5 across evaluation criteria, indicating that the approach is effective in providing supporting explanations for requirement ambiguities.

Paper C: Requirements Classification for Smart Allocation: A Case Study in the Railway Industry
Authors: *Sarmad Bashir*, Muhammad Abbas, Alessio Ferrari, Mehrdad Saadatmand, Pernilla Lindberg.
Status: Published in the 31st International Requirements Engineering Conference (RE 2023).
Abstract: Allocation of requirements to different teams is a typical preliminary task in large-scale system development projects. This critical activity is often performed manually and can benefit from automated requirements classification techniques. To date, limited evidence is available about the effectiveness of existing machine learning (ML) approaches for requirements classification in industrial cases. This paper aims to fill this gap by evaluating state-of-the-art language models and ML algorithms for classification in the railway industry. Since the interpretation of the results of ML systems is particularly relevant in the studied context, we also provide an information augmentation approach to complement the output of the ML-based classification. Our results show that the BERT uncased language model with the softmax classifier can allocate the requirements to different teams with a 76% F1 score when considering re-

quirements allocation to the most frequent teams. Information augmentation provides potentially useful indications in 76% of the cases. The results confirm that currently available techniques can be applied to real-world cases, thus enabling the first step for technology transfer of automated requirements classification. The study can be useful to practitioners operating in requirements-centered contexts such as railways, where accurate requirements classification becomes crucial for better allocation of requirements to various teams.

Paper D: ReqRAG: Enhancing Software Release Management through Retrieval-Augmented LLMs: An Industrial Study
Authors: Md Saleh Ibtasham, *Sarmad Bashir*, Muhammad Abbas, Zulqarnain Haider, Mehrdad Saadatmand, Antonio Cicchetti.
Status: Published in the 31st International Working Conference on Requirement Engineering: Foundation for Software Quality (REFSQ 2025).
Abstract: **[Context and Motivation]** Engineers often need to refer back to release notes, manuals, and system architecture documents to understand, modify, or upgrade functionalities in alignment with new software releases. This is crucial to ensure that new stakeholder requirements align with the existing system, maintaining compatibility and preventing integration issues. **[Problem]** In practice, the manual process of retrieving the relevant information from technical documentation is time-intensive and frequently results in inefficient software release management. **[Principal ideas/results]** In this paper, we propose a question-answering chatbot, *ReqRAG*, leveraging Retrieval Augmented Generation (RAG) with Large Language Models (LLMs) to deliver accurate and up-to-date information from technical documents in response to given queries. We employ various context retrieval techniques paired with state-of-the-art LLMs to evaluate the *ReqRAG* approach in industrial settings. Furthermore, we conduct human evaluations of the results in collaboration with experts from Alstom to gain practical insights. Our results indicate that, on average, 70% of the generated responses are adequate, useful, and relevant to the practitioners. **[Contribution]** Fewer studies have comprehensively evaluated RAG-based approaches in industrial settings. Therefore, this work provides technical considerations for domain-specific chatbots, guiding researchers and practitioners facing similar challenges.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Requirements in large-scale, complex industrial systems, such as in the railway domain, evolve through various stages and require coordinated efforts among stakeholders to ensure successful realization. However, the ongoing technological advancements coupled with prevalent manual practices in the requirements engineering (RE) process often lead to inefficiencies and scope creep. Accordingly, there is a need in industry for automated solutions that can augment and support the RE process by reducing practitioners' workload and enhancing efficiency. In this regard, the primary goal of this thesis relates to enhancing decision support for practitioners in requirements management for the development of complex industrial systems. To address this goal, we investigated the application of AI, specifically large language models (LLMs), to support and improve RE practices. In close collaboration with our industrial partner Alstom Rail AB (Alstom), we developed and validated these solutions.

The main contributions of the thesis include (i) automated extraction of requirements from tender documents *(Paper A, REQ-I)*, (ii) detection and explanation of requirements ambiguities *(Paper B, RADICLE)*, (iii) context-aware allocation of requirements to teams for implementation and testing (Paper C, *REQ-A*), and (iv) a domain-specific approach to retrieve requirements knowledge to facilitate the software development and release process (Paper D, *ReqRAG*). Internal evaluations of the deployed solutions, *REQ-I* and *REQ-A*, at

Alstom show that these tools can reduce manual effort by up to 80% on average compared to the manual baseline. On the other hand, the expert assessment of the results from *RADICLE* and *ReqRAG* indicates that both solutions can facilitate the practitioners in performing the RE tasks effectively.

5.2 Future Work

In future work, we plan to build upon the findings in this thesis to further support and advance RE practices, specifically for industries developing large-scale critical systems. Below, we outline possible future directions.

Understanding Requirements Evolution and Challenges in Industrial Contexts. Large-scale industries, such as railways, are characterized by complex and multi-faceted systems that involve various teams and stakeholders with different needs and perspectives within the RE domain. To gain practical insights, it is essential to investigate and document how requirements evolve during the development of large-scale systems and their related challenges. In this regard, we recently conducted a focus group study with practitioners from multiple companies, and a thematic analysis of the collected data is in progress. We intend to publish the results in the future.

Synthetic dataset generation for RE tasks. A consistent challenge in the RE domain is the scarcity of annotated datasets required for developing AI-based solutions, which rely on such data to learn domain-specific contexts effectively. In this regard, we plan to investigate the use of generative language models to produce synthetic datasets, particularly for RE tasks such as requirements classification. This will support the underrepresented classes in the annotated datasets and may improve the overall accuracy of the automated solutions. Furthermore, we plan to develop guidelines for generating effective synthetic datasets, which will be validated through expert reviews and applications in real-world settings to ensure quality and practical relevance.

Leveraging multimodal information for RE tasks. Technical documents used in RE, such as PDF tender documents and release notes, often include critical information in tables and images. Currently, in proposed solutions, such information is processed as plain text, disregarding the structural relationships and contextual information that is crucial for accurate interpretation. With the advent of recent multimodal language models, future work will focus on developing methods for effective extraction and representation of such structured and visual content. These methods will be evaluated to assess their performance compared to existing text flattening baselines.

Bibliography

[1] Maximilian Vierlboeck and Roshanak R Nilchiani. Requirement engineering in the age of system and product complexity–a literature review. In *2021 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–8. IEEE, 2021.

[2] Salih Sarp, Murat Kuzlu, Vukica Jovanovic, Zekeriya Polat, and Ozgur Guler. Digitalization of railway transportation through ai-powered services: digital twin trains. *European Transport Research Review*, 16(1):58, 2024.

[3] Luiz Eduardo G Martins and Tony Gorschek. Requirements engineering for safety-critical systems: A systematic literature review. *Information and software technology*, 75:71–89, 2016.

[4] Muhammad Abbas, Alessio Ferrari, Anas Shatnawi, Eduard Enoiu, Mehrdad Saadatmand, and Daniel Sundmark. On the relationship between similar requirements and similar software: A case study in the railway domain. *Requirements Engineering*, 28(1):23–47, 2023.

[5] Md Saleh Ibtasham, Sarmad Bashir, Muhammad Abbas, Zulqarnain Haider, Mehrdad Saadatmand, and Antonio Cicchetti. Reqrag: Enhancing software release management through retrieval-augmented llms: An industrial study. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 277–292. Springer, 2025.

[6] Iqtiar Md Siddique. Requirements engineering at a turning point: Key research trends shaping the future of systems and software integration. *Journal of Geotechnical Studies*, 9(3):10–46610, 2024.

[7] Fabiano Dalpiaz, Davide Dell'Anna, Fatma Basak Aydemir, and Sercan Çevikol. Requirements classification with interpretable machine learning and dependency parsing. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 142–152. IEEE, 2019.

[8] Zijad Kurtanović and Walid Maalej. Automatically classifying functional and non-functional requirements using supervised machine learning. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 490–495. Ieee, 2017.

[9] Anas Mahmoud and Grant Williams. Detecting, classifying, and tracing non-functional software requirements. *Requirements Engineering*, 21:357–381, 2016.

[10] Mehrdad Sabetzadeh and Chetan Arora. Practical guidelines for the selection and evaluation of natural language processing techniques in requirements engineering. In *Handbook on Natural Language Processing for Requirements Engineering*, pages 407–433. Springer, 2025.

[11] D Méndez Fernández, Stefan Wagner, Marcos Kalinowski, Michael Felderer, Priscilla Mafra, Antonio Vetrò, Tayana Conte, M-T Christiansson, Des Greer, Casper Lassenius, et al. Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice. *Empirical software engineering*, 22:2298–2338, 2017.

[12] Liping Zhao, Waad Alhoshan, Alessio Ferrari, Keletso J Letsholo, Muideen A Ajagbe, Erol-Valeriu Chioasca, and Riza T Batista-Navarro. Natural language processing for requirements engineering: A systematic mapping study. *ACM Computing Surveys (CSUR)*, 54(3):1–41, 2021.

[13] Syed Tauhid Ullah Shah, Mohammad Hussein, Ann Barcomb, and Mohammad Moshirpour. Explainability as a compliance requirement: What regulated industries need from ai tools for design artifact generation. *arXiv e-prints*, pages arXiv–2507, 2025.

[14] Yu Chen and J Jupp. Requirements engineering in complex infrastructure: Challenges to the development and management of rail transport requirements. In *Australasian Universities Building Education Association Conference*. Deakin University, 2021.

[15] Mack W Alford. A requirements engineering methodology for real-time processing requirements. *IEEE Transactions on Software Engineering*, (1):60–69, 1977.

[16] IEC ISO. Ieee. 29148: 2011-systems and software engineering-requirements engineering. Technical report, Technical report, 2011.

[17] Martin Glinz, Hans van Loenhoud, Stefan Staal, and Stan Bühne. Handbook for the cpre foundation level according to the ireb standard. *International Requirements Engineering Board*, 2020.

[18] Muhammad Abbas, Mehrdad Saadatmand, Eduard Enoiu, Daniel Sundamark, and Claes Lindskog. Automated reuse recommendation of product line assets based on natural language requirements. In *International Conference on Software and Software Reuse*, pages 173–189. Springer, 2020.

[19] Raffaele Cioffi, Marta Travaglioni, Giuseppina Piscitelli, Antonella Petrillo, and Fabio De Felice. Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. *Sustainability*, 12(2):492, 2020.

[20] Seemab Latif, Sarmad Bashir, Mir Muntasar Ali Agha, and Rabia Latif. Backward-forward sequence generative network for multiple lexical constraints. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 39–50. Springer, 2020.

[21] Alessio Ferrari and Andrea Esuli. An nlp approach for cross-domain ambiguity detection in requirements engineering. *Automated Software Engineering*, 26(3):559–598, 2019.

[22] Alessio Ferrari, Gloria Gori, Benedetta Rosadini, Iacopo Trotta, Stefano Bacherini, Alessandro Fantechi, and Stefania Gnesi. Detecting requirements defects with nlp patterns: an industrial experience in the railway domain. *Empirical Software Engineering*, 23:3684–3733, 2018.

[23] Sallam Abualhaija, Chetan Arora, Mehrdad Sabetzadeh, Lionel C Briand, and Michael Traynor. Automated demarcation of requirements in textual specifications: a machine learning-based approach. *Empirical Software Engineering*, 25:5454–5497, 2020.

[24] Jin Guo, Jinghui Cheng, and Jane Cleland-Huang. Semantically enhanced software traceability using deep learning techniques. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pages 3–14. IEEE, 2017.

[25] Jinfeng Lin, Yalin Liu, Qingkai Zeng, Meng Jiang, and Jane Cleland-Huang. Traceability transformed: Generating more accurate links with pre-trained bert models. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 324–335. IEEE, 2021.

[26] J Natt och Dag, Björn Regnell, Vincenzo Gervasi, and Sjaak Brinkkemper. A linguistic-engineering approach to large-scale requirements management. *IEEE software*, 22(1):32–39, 2005.

[27] Muhammad Abbas, Alessio Ferrari, Anas Shatnawi, Eduard Paul Enoiu, and Mehrdad Saadatmand. Is requirements similarity a good proxy for software similarity? an empirical investigation in industry. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 3–18. Springer, 2021.

[28] Waad Alhoshan, Liping Zhao, Alessio Ferrari, and Keletso J Letsholo. A zero-shot learning approach to classifying requirements: A preliminary study. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 52–59. Springer, 2022.

[29] Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*, 2014.

[30] Christopher D Manning. *Introduction to information retrieval*. Syngress Publishing,, 2008.

[31] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[32] Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *Transactions on Asian and Low-Resource Language Information Processing*, 20(5):1–35, 2021.

[33] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[34] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

[35] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[36] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[37] Sarmad Bashir, Muhammad Abbas, Mehrdad Saadatmand, Eduard Paul Enoiu, Markus Bohlin, and Pernilla Lindberg. Requirement or not, that is the question: A case from the railway industry. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 105–121. Springer, 2023.

[38] Sarmad Bashir, Muhammad Abbas, Alessio Ferrari, Mehrdad Saadatmand, and Pernilla Lindberg. Requirements classification for smart allocation: A case study in the railway industry. In *31st IEEE International Requirements Engineering Conference*, September 2023.

[39] Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[40] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[42] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[43] Saad Ezzini, Sallam Abualhaija, Chetan Arora, and Mehrdad Sabetzadeh. Automated handling of anaphoric ambiguity in requirements: a multi-solution study. In *Proceedings of the 44th international conference on software engineering*, pages 187–199, 2022.

[44] Tobias Hey, Jan Keim, and Sophie Corallo. Requirements classification for traceability link recovery. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 155–167. IEEE, 2024.

[45] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[46] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[47] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[48] Waad Alhoshan, Alessio Ferrari, and Liping Zhao. How effective are generative large language models in performing requirements classification? *arXiv preprint arXiv:2504.16768*, 2025.

[49] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.

[50] Sarmad Bashir, Alessio Ferrari, Abbas Khan, Per Erik Strandberg, Zulqarnain Haider, Mehrdad Saadatmand, and Markus Bohlin. Requirements ambiguity detection and explanation with llms: An industrial study. 2025.

[51] Lodiana Beqiri, Calkin Suero Montero, Antonio Cicchetti, and Andrey Kruglyak. Classifying ambiguous requirements: An explainable approach in railway industry. In *2024 IEEE 32nd International Requirements Engineering Conference Workshops (REW)*, pages 12–21. IEEE, 2024.

[52] Sarmad Bashir, Muhammad Abbas, Alessio Ferrari, Mehrdad Saadatmand, and Pernilla Lindberg. Requirements classification for smart allocation: A case study in the railway industry. In *2023 IEEE 31st International Requirements Engineering Conference (RE)*, pages 201–211. IEEE, 2023.

[53] Richard A Krueger. *Focus groups: A practical guide for applied research*. Sage publications, 2014.

[54] Gareth Terry, Nikki Hayfield, Victoria Clarke, Virginia Braun, et al. Thematic analysis. *The SAGE handbook of qualitative research in psychology*, 2(17-37):25, 2017.

[55] Seungjun Lee, Jungseob Lee, Hyeonseok Moon, Chanjun Park, Jaehyung Seo, Sugyeong Eo, Seonmin Koo, and Heuiseok Lim. A survey on evaluation metrics for machine translation. *Mathematics*, 11(4):1006, 2023.

[56] Hendrik Schuff, Lindsey Vanderlyn, Heike Adel, and Ngoc Thang Vu. How to do human evaluation: A brief introduction to user studies in nlp. *Natural Language Engineering*, 29(5):1199–1222, 2023.

[57] Sallam Abualhaija, Chetan Arora, Mehrdad Sabetzadeh, Lionel C Briand, and Michael Traynor. Automated demarcation of requirements in textual specifications: a machine learning-based approach. *Empirical Software Engineering*, 25:5454–5497, 2020.

[58] Abhishek Sainani, Preethu Rose Anish, Vivek Joshi, and Smita Ghaisas. Extracting and classifying requirements from software engineering con-

tracts. In *2020 IEEE 28th international requirements engineering conference (RE)*, pages 147–157. IEEE, 2020.

[59] Jonas Winkler and Andreas Vogelsang. Automatic classification of requirements based on convolutional neural networks. In *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, pages 39–45. IEEE, 2016.

[60] Andreas Falkner, Cristina Palomares, Xavier Franch, Gottfried Schenner, Pablo Aznar, and Alexander Schoerghuber. Identifying requirements in requests for proposal: A research preview. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 176–182. Springer, 2019.

[61] Rajni Jindal, Ruchika Malhotra, and Abha Jain. Automated classification of security requirements. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2027–2033. IEEE, 2016.

[62] Vasily Varenov and Aydar Gabdrahmanov. Security requirements classification into groups using nlp transformers. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pages 444–450. IEEE, 2021.

[63] Henning Femmer, Daniel Méndez Fernández, Stefan Wagner, and Sebastian Eder. Rapid quality assurance with requirements smells. *Journal of Systems and Software*, 123:190–213, 2017.

[64] A Handbook. From contract drafting to software specification: Linguistic sources of ambiguity. 2003.

[65] Vincenzo Gervasi, Alessio Ferrari, Didar Zowghi, and Paola Spoletini. Ambiguity in requirements engineering: Towards a unifying framework. *From Software Engineering to Formal Methods and Tools, and Back: Essays Dedicated to Stefania Gnesi on the Occasion of Her 65th Birthday*, pages 191–210, 2019.

[66] Antonio Bucchiarone, Stefania Gnesi, Giuseppe Lami, Gianluca Trentanni, and Alessandro Fantechi. Quars express-a tool demonstra-

tion. In *2008 23rd IEEE/ACM International Conference on Automated Software Engineering*, pages 473–474. IEEE, 2008.

[67] Sri Fatimah Tjong and Daniel M Berry. The design of sree—a prototype potential ambiguity finder for requirements specifications and lessons learned. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 80–95. Springer, 2013.

[68] Alessio Ferrari, Giuseppe Lipari, Stefania Gnesi, and Giorgio O Spagnolo. Pragmatic ambiguity detection in natural language requirements. In *2014 IEEE 1st International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 1–8. IEEE, 2014.

[69] Alessio Ferrari and Stefania Gnesi. Using collective intelligence to detect pragmatic ambiguities. In *2012 20th IEEE International Requirements Engineering Conference (RE)*, pages 191–200. IEEE, 2012.

[70] Alvaro Veizaga, Seung Yeob Shin, and Lionel C Briand. Automated smell detection and recommendation in natural language requirements. *IEEE Transactions on Software Engineering*, 50(4):695–720, 2024.

[71] Hui Yang, Anne De Roeck, Vincenzo Gervasi, Alistair Willis, and Bashar Nuseibeh. Analysing anaphoric ambiguity in natural language requirements. *Requirements engineering*, 16:163–189, 2011.

[72] Savas Yildirim, Garima Malik, Mucahit Cevik, and Ayşe Başar. Anaphora resolution in software requirements engineering: A comparison of generative NLP pipelines and encoder-based models. In *2024 34th International Conference on Collaborative Advances in Software and COmputiNg (CASCON)*, pages 1–6. IEEE, 2024.

[73] Jane Cleland-Huang, Raffaella Settimi, Xuchang Zou, and Peter Solc. Automated classification of non-functional requirements. *Requirements engineering*, 12(2):103–120, 2007.

[74] Jane Cleland Huang, Raffaella Settimi, Xuchang Zou, and Peter Solc. The detection and classification of non-functional requirements with application to early aspects. In *14th IEEE International Requirements Engineering Conference (RE'06)*, pages 39–48. IEEE, 2006.

[75] Jane Cleland-Huang, Sepideh Mazrouee, Huang Liguo, and Dan Port. NFR, March 2007.

[76] Raul Navarro-Almanza, Reyes Juarez-Ramirez, and Guillermo Licea. Towards supporting software engineering using deep learning: A case of software requirements classification. In *2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT)*, pages 116–120. IEEE, 2017.

[77] Osamah AlDhafer, Irfan Ahmad, and Sajjad Mahmood. An end-to-end deep learning system for requirements classification using recurrent neural networks. *Information and Software Technology*, 147:106877, 2022.

[78] Tobias Hey, Jan Keim, Anne Koziolek, and Walter F Tichy. Norbert: Transfer learning for requirements classification. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 169–179. IEEE, 2020.

[79] Ethem Utku Aktas and Cemal Yilmaz. Automated issue assignment: results and insights from an industrial case. *Empirical Software Engineering*, 25(5):3544–3589, 2020.

[80] Leif Jonsson, Markus Borg, David Broman, Kristian Sandahl, Sigrid Eldh, and Per Runeson. Automated bug assignment: Ensemble-based machine learning in large scale industrial contexts. *Empirical Software Engineering*, 21(4):1533–1578, 2016.

[81] Muhammad Abbas, Sarmad Bashir, Mehrdad Saadatmand, Eduard Paul Enoiu, and Daniel Sundmark. Requirements similarity and retrieval. In *Handbook on Natural Language Processing for Requirements Engineering*, pages 61–88. Springer, 2025.

[82] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

[83] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms:

Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501, 2024.

[84] Syed Juned Ali, Varun Naganathan, and Dominik Bork. Establishing traceability between natural language requirements and software artifacts by combining rag and llms. In *International Conference on Conceptual Modeling*, pages 295–314. Springer, 2024.

[85] Saad Ezzini, Sallam Abualhaija, Chetan Arora, and Mehrdad Sabetzadeh. Ai-based question answering assistance for analyzing natural-language requirements. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 1277–1289. IEEE, 2023.

[86] Chetan Arora, Tomas Herda, and Verena Homm. Generating test scenarios from nl requirements using retrieval-augmented llms: An industrial study. *arXiv preprint arXiv:2404.12772*, 2024.

[87] Daksh Chaudhary, Sri Lakshmi Vadlamani, Dimple Thomas, Shiva Nejati, and Mehrdad Sabetzadeh. Developing a llama-based chatbot for ci/cd question answering: A case study at ericsson. *arXiv preprint arXiv:2408.09277*, 2024.

[88] Kari Lukka. The constructive research approach. *Case study research in logistics. Publications of the Turku School of Economics and Business Administration, Series B*, 1(2003):83–101, 2003.

[89] Sarmad Bashir. Towards ai-centric requirements engineering for industrial systems. In *46th International Conference on Software Engineering: Companion Proceedings*, April 2024.

[90] Youguo Li and Haiyan Wu. A clustering method based on k-means algorithm. *Physics Procedia*, 25:1104–1109, 2012.