CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Enhancing feasibility analysis through LLMs: An Action Research Approach

Investigating the RAG architecture and finetuned LLMs as a tool for decision-support in feasibility analysis

Master's thesis in Software Engineering and Technology

David Johansson
Hampus Jansson

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Master's thesis 2025

Enhancing feasibility analysis through LLMs: An Action Research Approach

Investigating the RAG architecture and finetuned LLMs as a tool for decision-support in feasibility analysis

David Johansson
Hampus Jansson

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

Investigating the RAG architecture and finetuned LLMs as a tool for decision-support in feasibility analysis
NAME FAMILYNAME

Supervisor: Khan Mohammad Habibullah, Chalmers University of Technology
Advisor: Andreas van der Weide, Siemens Energy AB
Examiner: Irum Inayat, Department of Computer Science and Engineering

Master's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice Gothenburg, Sweden 2025

Enhancing feasibility analysis through LLMs: An Action Research Approach
Investigating the RAG architecture and finetuned LLMs as a tool for decision-support in feasibility analysis
HAMPUS JANSSON
DAVID JOHANSSON
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

Requirement elicitation is a critical part of Requirements engineering (RE). However, it is heavily reliant on the knowledge of domain experts, is prone to human-errors, and remains a time-consuming process. This study investigates how Large Language Models can assist domain experts in conducting feasibility analyses of customer demands. Specifically, the study investigates two different solutions as an LLM; the Retrieval-Augmented Generation (RAG) architecture, and the finetuned pre-trained LLM. The two models were developed using the action research methodology in collaboration with Siemens Energy AB. All the data used for training the models were related to historical gas turbine projects and were provided by Siemens Energy. The findings showed that the finetuned LLM struggled with ambiguous requirements and was prone to hallucinate for a certain type of customer demands, in particular demands that were covered by Siemens Energy's standard. In contrast, the RAG demonstrated a higher accuracy and relevance in its outputs. The two models were evaluated through a survey, which was answered by the domain experts at Siemens Energy. The surveys revealed that both of the models have potential as a decision-support tool, but that the RAG was preferred since it outperformed the finetuned LLM in all of the metrics. Lastly, a finetuned embedding model was developed as part of the RAG solution. This embedding model was quantitatively evaluated and compared to state-of-the-art models. This evaluation showed that the fine-tuned model outperforms state-of-the-art benchmarks on the intended task and in the environment of Siemens Energy.

Keywords: Requirements Engineering, Feasibility Analysis, Large Language Models, RAG, Finetuning, NLP, Action Research, Siemens Energy.

Acknowledgements

We want to thank our supervisor at Chalmers, Khan Mohammed Habibullah, and Andreas van Der Weide, for their support and guidance during the project.

All data and computational resources were provided by Siemens Energy AB.

Hampus Jansson & David Johansson, Gothenburg, May 2025

Contents

List of Figures

List of Tables

1
Introduction

Requirements Engineering (RE) is a critical step in the field of software engineering, ensuring that a product's requirements agree with business objectives, technical constraints, and customer's needs [1]. One essential part of RE is gathering requirements, known as the requirements elicitation. This phase plays a significant part in a project's success. Research has shown that systems failure can be traced back to poor requirements elicitation in up to 90% of large software projects [2]. An important step in elicitation is the feasibility analysis, which assesses whether the specific wants from the customer is viable within the constraints. Constraints such as time, budget, and technical limitations while meeting the customer's needs [3]. The feasibility analysis ensures that impractical or problematic requirements are identified early rather than later in the process, reducing the risk of mistakes that could be costly to correct [4].

As of today, feasibility analysis are done manually by experts within the domain of the product, thus requiring large efforts to analyze customers requirements and compare these with the capabilities of the product offered [5][6]. This process is time consuming and prone to human errors, since it includes constantly comparing the customer's requirements with the capabilities of the product or system offered [6]. Experts must thoroughly assess each requirement, leading to inconsistencies due to subjective interpretations. Ensuring understanding between the stakeholders, in this case the customer and the seller, is crucial for RE. Mistakes made in this step is often not found until later down the line and can often be costly to fix. A study conducted by Mogyorodi [7] showed that 56% of bugs in a project was related to faulty requirements. Customer requirements are typically written in natural language, which introduces ambiguities and inconsistencies [4]. The vagueness of natural language makes it difficult for humans to ensure uniform understanding, further complicating the feasibility analysis.

AI tools offer a promising solution to reduce the workload of experts in the feasibility analysis process by assisting with requirement analysis and validation [8]. Unlike full automation, AI can act as a tool to assist in decision-making, helping experts quickly identify infeasible or ambiguous requirements, thereby improving efficiency and reducing the risk of overlooking constraints. While there has not been much research on integrating AI into the process of feasibility analysis specifically, the application of AI solutions in RE as a whole is becoming much more prevalent [8][9][10]. Leveraging NLP and ML, AI tools can help structuring requirements, check for inconsistencies, summarizations or act as conversational agents during

product development. Thus, AI-driven tools can enhance, rather than replace, human expertise by streamlining processes and reducing manual effort, resulting in more robust processes.

Recent advances in NLP have made LLMs and RAG architectures increasingly useful techniques for tackling complex language tasks [11]. Finetuned LLMs build on the inherent structure of pre-trained models by training them on domain-specific data [12]. This allows the LLM to better capture context relevant to the tasks such as interpreting requirements. RAGs, on the other hand, enhance an LLM's performance by leveraging an embedding model to retrieve relevant context from a knowledge base for the LLM to use [13]. This architecture is particularly applicable in tasks where up-to-date and domain-specific information is important for generating a good response. Crucially, what both solutions offer, is an ability to generate output in natural language that closely resembles human reasoning [11]. This makes LLMs highly attractive for practical use.

This study aims to investigate how LLMs can support experts in conducting feasibility analysis more efficiently. Specifically, this study explores the integration of two specific LLM solutions. It will be investigated how a fine-tuned pre-trained LLM and an RAG can assist in comparing customer requirements against product specification, flag potential issues, and generate structured feedback. The results will be evaluated through both quantitative and qualitative metrics, which will be obtained by surveying domain experts. The research will utilize data provided by Siemens Energy AB on their products, specifically focused on gas turbines, as well as their historical client requirements. The dataset contains past customer specification sheets and the corresponding comments on the identified faulty requirements. Access to this dataset, along with the help and feedback from the experts at Siemens Energy, offers an opportunity to ensure that the solutions are both practical and relevant to real-world applications. By combining the latest research of LLMs with the expertise of experts, the study aims to advance the research field of adopting LLMs in the process of feasibility analysis.

The research questions (RQs) that this study aims to answer are as follows:

RQ1 *How does Large Language Models (LLM) perform in assisting domain experts in the feasibility analysis?*

RQ1.1 *How does a finetuned pre-trained LLM perform in assisting domain experts in the feasibility analysis?* The approach will be evaluated quantitatively and qualitatively in terms of the following metrics:

- Relevance: How relevant is the output to the demand?
- Correctness: How correct and factual is the information in the output?
- Assistance: How helpful is the output in determining the feasibility of the demand?

The evaluating will be done by experts involved in the feasibility analysis at the energy corporation.

RQ1.2 *How does a Retrieval Augmented Generation (RAG) model perform in assisting domain experts in the feasibility analysis?* The approach will be evaluated quantitatively and qualitatively in terms of the following metrics:

- Relevance: How relevant is the output to the demand?
- Correctness: How correct and factual is the information in the output?
- Assistance: How helpful is the output in determining the feasibility of the demand?

The evaluating will be done by experts involved in the feasibility analysis at the energy corporation.

RQ1.2.1 *How does the performance of a finetuned embedding model compare to state-of-the-art models, such as OpenAIs text embedding models, in the context of information retrieval tasks?* The comparison will be evaluated quantitatively using metrics such as accuracy and precision, specifically Recall@K and Mean Ranking Reciprocal (MRR).

2
Background

This chapter explores the current practices in RE, focusing on the elicitation and specifically feasibility analysis. It also dvelves into the state-of-the-art for NLP tasks with emphasis on LLMs, RAG and Embedding models. Each section provides an overview of these models and their impact on the field of NLP.

2.1 Requirements Engineering (RE)

RE is a critical part of software and systems development [1]. Its purpose is identifying, analyzing, specifying, and validating the needs and constraints of stakeholders, ensuring that the final product aligns with the business objective, technical limitations and user expectations. The quality of RE has a direct impact on the overall success of a project, and its failure often leads to time delays, overspent budget, and a project failure [2].

2.1.1 Requirements Elicitation

One important step within RE is requirements elicitation, which involves gathering requirements from the stakeholders to understand what the system needs to be able to do [2]. Elicitation can be challenging due to the subjective nature of stakeholder input, incomplete or ambiguous descriptions, and the complexity of translating business needs into requirements. It is not just a matter of collecting information. The process requires critical thinking, domain expertise and clear communication between stakeholders. If an error occurs in the elicitation phase, then it will be carried on to the further stages which will potentially damage the whole product.

Elicitation typically includes a combination of interviews, workshops, observations, and prototyping [14]. However, regardless of the method, the success of elicitation depends on the clarity and feasibility of the requirements obtained. This leads to the vital task if assessing each requirements early on, which is where the feasibility analysis plays an important role [2].

2.1.2 Feasibility Analysis

A feasibility analysis is used to justify a project [15]. It compares various requirements based on their economic, technical and operational feasibility. Such constraints can be time, budget, or technical limitations. In addition, it also ensures

that stakeholder expectations are aligned with what the organization and system can deliver [2].

In practice, feasibility analyses are carried out manually by domain experts who must iteratively compare customer demands with the product's technical specification and organization standards [6]. The process is prone to inconsistencies due to varying interpretations of natural language requirements.

In more complex engineering domains such as energy systems, feasibility becomes particularly demanding. Requirements are often embedded in lengthy, unstructured documents and described in certain terminology specific to the domain. Misinterpretations or undetected infeasible demands at this stage can prove to be costly later down the line, leading to time and cost implications during implementation [7].

2.2 Large Language Models (LLMs)

Recent advancements in Natural Language Processing (NLP), particularly in the developments of LLMs, have motivated researchers in RE to explore the potential advantages of these tools in enhancing RE tasks [16]. LLMs are trained on a large amount of data to be able to both understand as well as generate natural language. LLMs are typically based on the Transformer architecture [17]. An important feature of Transformers is the self-attention mechanism. This mechanism allows the LLM to weigh the importance of different words in a sentence when understanding and generating the text. Moreover, it enables the model to capture more complex linguistic patterns and deeper context. This section explains the evolution of LLMs, finetuning, their integration with RAG architectures, and embedding models role of allowing semantic understanding and similarity measurement.

2.2.1 Finetuned Pre-Trained Large Language Models

Pre-trained LLMs, such as Mistral [18], Gemma [19], and LLaMA [20], have become increasingly prominent in the field of NLP due to their ability to generalize across diverse unseen tasks [21]. They are trained on large corpora, enabling them to develop a broad understanding of language. However, while they perform well on general tasks, finetuning is most often needed to improve their performance for specific applications.

Finetuning involves adapting a pre-trained model for a particular task or domain by training on a smaller dataset specific to the task. This process leverage the knowledge already embedded in the model, making it more efficient than building a model from scratch. Three keys methods for finetuning pre-trained LLMs are:

Transfer Learning, where a pretrained model is further trained on task-specific data. For example a model trained on general text can be finetuned on product specifications to assist in helping developers find information that can otherwise be hard to find. Transfer learning allows the model to adapt its general language un-

derstanding to the target domain. [21]

In-Context Learning, which involves providing the model with examples of input-ouput pairs during inference, without modifying the model's parameters. This method allows the model to learn tailored responses for the task by providing it with prompts that includes the input-output examples. In-Context learning is especially useful for tasks where labeled data is scarce or when fast prototyping is required. [22]

Prompt Learning, is a technique which uses carefully selected prompts to guide a model to produce certain results or behave in a specific way. It is common to adopt prompt learning with pre-trained LLMs in order to achieve a certain behavior and to guide it in the style in which it answers. Hence, this approach is of particular relevance when the model is used in very specific domains. [23]

Still, conventional fine-tuning large models comes with a high computational cost, since it means retraining all model parameters. Researchers at Microsoft [24] came up with an interesting solution to this. They presented Low-Rank Adaption (LoRA), this approach freezes the pre-trained model weights and instead injects trainable rank decomposition matrices into each layer of the Transformer architecture. This approach leads to less trainable parameters for down-stream tasks. Further, research by Dettmers et al. [12] presented an efficient fine-tuning approach called QLoRA that followed up on LoRA. QLoRA backpropagates gradients through a frozen, 4-bit quantized pre-trained language model into LoRA. It allows the pre-trained model to keep its full task-performance.

2.2.2 Retrieval Augmented Generation (RAG)

A RAG architecture combines the strength of retrieval-based and generation-based models to advance the performance of NLP tasks [13]. RAG is especially useful for tasks that requires generating information and contextually relevant responses, such as question answering, conversational agents and summarizations. The retriever is responsible for retrieving contextually relevant documents, or chunks of text, from a large corpus based on the input query. The generator then uses the retrieved information to generate the response. This combination allows RAG models to leverage external knowledge, not stored in the model itself, making it more robust and capable of handling different types of queries.

The RAG model typically uses an embedding model to encode both the query and the external knowledge corpus. The corpus most often consists of documents that have been split into chunks, or paragraphs, of text. From the query, the retrieval model can find the most semantically similar text chunks. These chunks are thereafter used as the expanded context in a prompt posed to an LLM. The model's approach to answer depends on the task-specific criteria, allowing it to either draw upon its inherent parametric knowledge or restrict its response to the information provided by the context. [25]

2.2.3 Embedding Models

Embedding models are a crucial part of NLP, especially for LLMs. They take textual data, or human language, and transform it into numerical representations, which enables machines to process and understand it. Embedding models can capture semantic relationships between words, phrases and sentences, which can help in various NLP tasks such as text classification, sentence similarity and machine translation. [26]

2.2.3.1 Word Embeddings

Word embedding are dense vector representations in a multi-dimensional space, where the distance and direction between vectors reflects the similarity and relationships between the words. The concept of word embeddings gained traction with the introduction of models like Word2Vec [27] and GloVe [28]. Word2Vec uses two architectures: Continuous Bag of Words (CBOW) and Skip-gram [27]. CBOW predicts a word based on the context, while Skip-gram predicts the context based on the word. GloVe constructs word vectors leveraging global word-word co-occurrence statistics from a corpus [28]. The advantages of word embeddings are many. Firstly, the models can capture the semantic similarity between two words. For example, the vectors of “king“ and “queen“ are close to each other in the vector space compared to the vector of “car“. Embeddings can also capture analogies, such as “king-man+woman≈queen“. Lastly, compared to traditional methods such as one hot encoding [29], the embeddings greatly reduce dimensionality of textual data, making it much more manageable for machines to process.

2.2.3.2 Sentence Embeddings

While word embeddings capture word-level semantics, they fall short in capturing the semantics at a sentence-level. Sentence embeddings address this issue by encoding the sentences into fixed-length vectors to capture the meaning of the sentence. There are many proposed methods for generating sentence embeddings. A simple approach is to average the word embeddings of a sentence. However, this method often fails to capture all contextual information. State-of-the-art models like BERT [30], Sentence-BERT [31] and Universal Studio Encoder [32] all uses the Transformer architecture which has revolutionized the field of NLP through enabling parallel processing of sentences.

2.2.3.3 Sentence Similarity

Sentence Similarity is a NLP task that takes the meaning of two snippets of text and assigns a similarity score to them [33]. Existing methods to measure sentence similarity face two main changes: (1) labeled datasets are often limited in size, making it hard to train supervised neural models; and (2) unsupervised LMs are typically learned to understand text on a phrase or word level which results in a gap in what they were trained to do (phrase-level or word-level understanding) and what they are tested on (sentence-level semantics) [34]. The most used methods for

evaluating similarity is cosine similarity, Euclidean distance and Manhattan distance [35].

3

Related Works

3.1 Requirements Engineering and Elicitation challenges

RE has been intensively studied in the last decade [36]. However, it still remains one of the most critical processes in software development, with recent studies showing that 56% of system failures comes from faulty requirements. Besrour et al. [36] conducted a literature review where they identified a total of 24 RE challenges, with 12 of them being classified critical. Among these, "Collecting vague and ambiguous requirements" were the most critical based on a support scale, with "Undocumented Functional requirements and non-functional requirements" and "Requirements inconsistency" being a close third and second.

In Kauppinen et al. [37], the authors highlights how implementing an RE process is challenging and complex assignment. They discovered that the success of an RE process implementation greatly depends on human factors, such as motivation, commitment, and enthusiasm. The authors emphasizes how factors, such as the definition of the RE process scope, provide adequate training and support resources, reducing the duration of executing a new RE process, and defining proper measurement tools, are crucial for successful implementation.

Ambiguity in natural language requirements have for a long time been a recognized as an inevitable challenge in RE [38]. Muneera Bano [38] conducted a mapping study focusing on NLP techniques to address the ambiguity in requirements. The author concluded that the RE research community has mainly focused on detecting of ambiguity, while avoiding or resolving ambiguity has been largely neglected.

To determine the strengths and the weaknesses of different elicitation techniques based the challenges faced by RE engineers, Okesola et al. [39] conducted qualitative comparison of eight different different elicitation methods: Brainstorming, Workshops, Prototyping, Joint application, Group work, Ethnography, Introspection and User scenarios. The authors compared the methods using two criteria: (1) the quality of feedback, which includes proximity to use, effort per user, and required skill, and (2) the terms of the collection of information, which includes structure, richness and quantifiability. The comparison showed that each one of the eight elicitation techniques has its strengths and weaknesses that has to be considered when choosing which method to use.

Lui et al. [40] conducted a questionnaire survey to identify the main reasons of failure in RE practices, with an emphasis on requirements elicitation approaches and requirements representation techniques. The authors identified eight main challenges, with three directly linked to communication between stakeholders, namely "Customers do not have a clear understanding of system requirements themselves", "Broken communication links between customer, analyst and developer", and "Users' needs and understanding constantly change". Further, two of the challenges found were caused by insufficient domain expertise "Reuse existing design in wrong context and environment" and "Requirements decision-makers lack of technical and domain expertise".

3.2 Approaches to Feasibility Analysis

Feasibility analysis is a crucial step in evaluating whether customer requirements can be realistically implemented given constraints such as technical limitation, budget, and time [15]. Traditionally, this process has been done manually and been reliant on the expertise of domain professionals [15][5]. Pergl [15] defines the feasibility study as consisting of the following dimensions: economic feasibility, technical feasibility, and operational feasibility. All of which must be assessed properly early in the RE process to prevent issues later on.

Manual feasibility analysis, while thorough, has several limitations. It is time-consuming and prone to subjective interpretations, especially when working with ambiguous or poorly formulated natural language requirements [4][7]. Necula et al. [6] conducted a systematic literature review where they found that the majority of RE tasks, feasibility analysis included, rely solely on expert judgment rather than formalized or automated methods. Furthermore, they highlight the lack of tools for specific RE tasks in the industry.

Sam McLead [41] conducted an integrative review enhancing the approach to feasibility studies for novel, complex, or unfamiliar projects. The author found while evaluating feasibility is a often considered a narrow technical question, it relies upon on examination of various interlinked factors that contribute to long-term, life cycle success. McLead outlines seven principles developed in the study: (1) going beyond singular questions to assess a proposal from many perspectives, (2) understanding and reducing forms of uncertainty through iterative research questions, (3) drawing upon many perspectives and inputs, (4) situating a proposal within a reference set of other existent and failed examples, (5) critically assessing organizational capabilities, (6) reporting findings in a consistent way, and (7) view a proposal across a longer-term time horizon.

Sai Ganesh Gunda [3] published a research paper focused on comparing different methods for the requirements gathering process, such as feasibility analysis. The authors sampled 35 RE professionals to conduct a survey. From the results of the survey it was concluded that when using various requirements gathering method

that “experience matters“. Experienced RE professionals better understand the nuances of various methods make more informed decisions, while less experienced RE professionals might not.

3.3 Automation in RE

In a market research analysis on requirements analysis by by Luisa et al. [42], a survey was conducted to determine the potential demand for a computer-aided software engineering tool. From the survey, it was found that 69% of the respondents considered automation is the most valuable contribution to the field of requirements elicitation. Wong et al. [43] builds on this finding, highlighting automation as is at the top of the wish list of most software developers.

Ronit Ankori [44] presents a new method for automatically retrieving functional requirements from the stakeholders. The author highlights how knowing what it is we want to develop and correctly defining it plays a crucial role in the software engineering field. The tools aim is to collect information from identified stakeholders and then manipulate this data into finalized requirements. Using automation, the author argues his tool ease the requirements elicitation process.

3.4 LLMs in RE

Recently, there has been emerging more and more research papers on integrating machine learning into various RE related tasks, especially utilizing LLMs [45]. Ronanki et al. [45] investigates the use of pre-trained LLMs and prompt engineering for the intention of semi-automating specific RE tasks. Through their findings the authors highlight that while LLMs with specific prompt patterns could improve the process of different RE activities, the use of LLMs in RE tasks should be limited to assisting relevant RE stakeholders with appropriate human oversight in place of automating these tasks.

ChatGPT has gained significantly more recognition due to its notable improved performance in NLP tasks [46]. As a result Ronanki et al. [46] chose to investigate the potential of ChatGPT to assist in requirements elicitation processes. The authors posed both ChatGPT and five RE experts with six questions to elicit requirements. The results showed how ChatGPT-generated requirements are considered highly abstract, atomic, consistent, correct and understandable in comparison to human RE experts’ formulated requirements. Ronanki et al. underscores the importance for the RE research community to further investigate ChatGPT’s and other LLMs’ use cases in various RE tasks.

Many applications in NLP rely on adapting large scale pre-trained LLM to downstream applications using finetuning [24]. VM et al. [47] address the challenges of finetuning pre-trained LLMs for enterprises through providing a guide for how to finetune LLaMa, an open source LLM. The authors highlight how quantization and

Low Rank Adaption (LORA) finetuning of LLMs has opened up a whole new possibility of finetuning LLMs on smaller GPUs and domain-specific datasets. However, the results still show how the LLM is prone to hallucinations. To mitigate this, the authors argue that the developer can experiment with prompt engineering and use data of high-quality.

Narmani et al. [48] investigated how integrating RAG could improve the troubleshooting real-time requirements of industrial environments. The authors highlight how the traditional manual approach rely heavily on human expertise and static documentation. The findings showed that the integration of a RAG into the process of troubleshooting consequentially leads to substantial improvement in terms of operational efficiency and complex problem-solving. This was due to the RAGs ability to quickly retrieve relevant information from a vast amount of data combined with the generative AI in the end of the pipeline, which meant it could address complex issues in terms of accuracy, relevance and time.

Masoudifard et al. [16] argues that ensuring that Requirements Specifications align with higher level organizational or national requirements is vital. The authors further demonstrates that integrating a robust Graph-RAG network in combination with prompt engineering techniques can significantly enhance the performance of specific RE tasks. However, while the approach gives more accurate and context-aware results compared to baseline RAG methods, it is both costly and more complex to implement. Additionally, its effectiveness heavily relies on high-quality data, which the authors point out is most often not feasible in industrial settings.

4
Research Methodology

To investigate our research questions, this study adopts the Action Research methodology. Action research is a well suited methodology for industry-academia collaborations in software engineering and helps enhance the research's relevance [49]. Furthermore, it can assist in creating innovative solutions and helps gain in-depth knowledge of new development in real-world scenarios. The method is especially fitting for projects where problem-solving and knowledge collection occur simultaneously. This was the case at Siemens Energy AB, as they had no clear idea how the optimal solution would look or how it would be developed.

We chose to follow the five steps in the action research cycle presented by Miroslaw Staron [50], as they have been recognized to work well in action research approaches. These steps, as seen in figure 4.1, are:

- Diagnosing: The researchers identifies and examines the problems with feasibility analysis through literature review, observations, and engagement from RE practitioners at Siemens Energy.
- Action planning: After diagnosing, the researchers in collaboration with the practitioners at Siemens Energy develop an action plan. This plan outlines experiments and objectives.
- Action taking: In this phase, the planned experiments and actions are implemented. The researchers and stakeholders collaborate to execute the implementations.
- Evaluation: The implemented actions are evaluated.
- Specifying learning: This phase is about gathering the outcomes and experiences gained from the action research process and analyze how they can be applied to the next cycle.

In our study, we went through this cycle twice.

4.1 Context

We conducted the action research study in collaboration with Siemens Energy AB. The corporation is active across the entire energy landscape. Their mission is to support the world's needs to reduce greenhouse gas emissions and make energy more reliable, affordable, and sustainable. Siemens Energy is focused on developing, manufacturing, and servicing of gas turbines for power generation, compressor drives, as well as heat pumps. It is together with a team of project lead engineers at Siemens Energy, responsible for customizing the gas turbines to satisfy the customer's needs

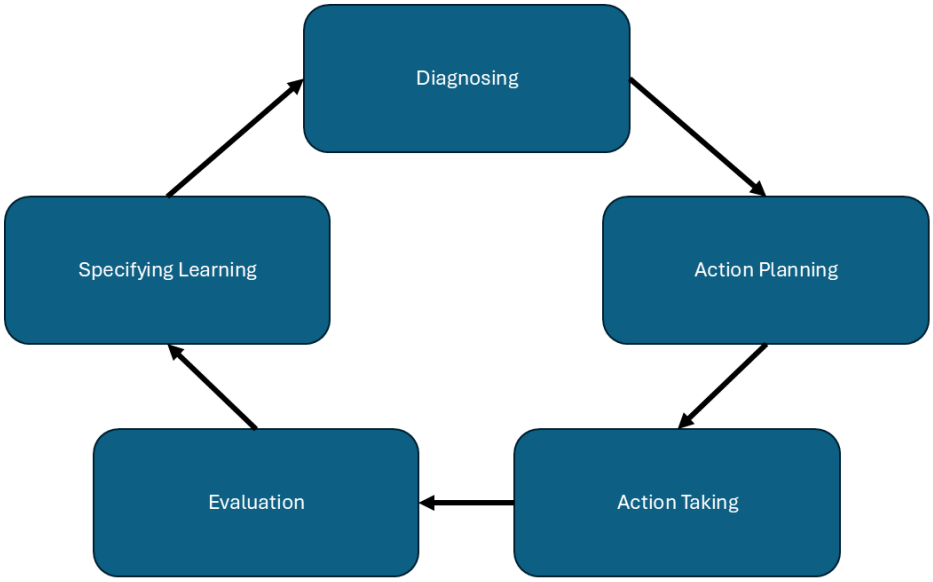


Figure 4.1: Action Research Model

while leading a team of engineers to adapt to the new constraints, that we are conducting the research.

We began our research by engaging with the experts within RE for gas turbines at Siemens Energy, who presented the problems that came with manual feasibility analysis in RE for the purpose of requirements elicitation at their organization. The initial discussions highlighted two main challenges: (1) subjective interpretations and the ambitiousness of customer demands often leads to human errors, and (2) the feasibility study being a time consuming process, requiring multiple experts to analyze the customer demands and iteratively compare with the product's specifications. To further build an understanding of the problem, we conducted an extensive review of existing literature and research, which confirmed that this is an issue prevalent in the industry as a whole. From the insights gathered during this phase, we derived three key objectives: (1) the need for efficiency, (2) accuracy, and (3) consistency in the feasibility analysis.

As of now, Siemens Energy's experts works in a software program where they can mark certain sections in the customer's specification sheet and provide a comment. The sections that are marked and commented are most often sections containing infeasible or problematic requirements. This process of finding infeasible requirements includes iteratively comparing all customer requirement with the gas turbine's specifications to ensure that no faulty requirements are overlooked. As a result of this, Siemens Energy is looking for a way to integrate AI as a tool in this process for the experts to use to reduce the workload and increase accuracy of the task.

The action research team that investigated the proposed research questions consisted of two researchers, four practitioners and two stakeholders. The researchers

and practitioners collaborated in the collection of data for the research. The two researchers and one practitioner formed the core development team. The project stakeholders included Chalmers University of Technology as well as Siemens Energy.

Siemens Energy did not have a proposed approach to solve the problems in the process of feasibility analysis, but only expressed the desire to integrate AI into the process. We proposed an AI-tool that can work as decision-support for the experts, reducing the amount of manual labor and improve accuracy of the task. Siemens Energy agreed on this idea, and we started to investigate different solutions to realize the tool. To select our initial approach we developed multiple non-functional prototypes, meaning prototypes that can perform small parts of the task but wont work in a real-world scenario. This was to visualize how a solution could look like. Among these prototypes were finetuned pre-trained LLMs from huggingface, a BERT classifier and a RAG model with a finetuned sentence transformer, all of which were trained on a smaller subset of data. The prototypes were subsequently discussed with the practitioners at Siemens Energy. From these discussions it could be determined that an LLM type response was the best fit for their organization. Therefore, the continued research focused on using an finetuned pre-trained LLM or a RAG as the solution.

4.2 Action Research Cycle 1: Finetuned pre-trained LLM

The first cycle of our action research focused on fine-tuning a pre-trained LLM, as earlier works highlighted its potential in finetuning for domain-specific datasets. The primary objectives were to overcome hardware limitations and address the scarcity of high-quality data for the task at hand. By collaborating with practitioners at Siemens Energy, the research sought to develop a solution that could be integrated into their existing workflow.

4.2.1 Diagnosis

In the diagnosis phase of Cycle 1, the researchers investigated the use of a fine-tuned pre-trained LLM to answer RQ1. Two main limitations were identified for this approach: (1) the capabilities of the hardware available to the research would pose some limitations on model selection and model training, and (2) the scarcity of high-quality data for this task had to be addressed. The practitioners at Siemens Energy provided an extensive list of historical faulty customer requirements paired with a comment on its feasibility by experts at Siemens Energy. However, many of the requirement had not been properly extracted. This happens when the expert working on the project only marks a header or a keyword from the customer specification sheet instead of the complete requirement when providing their comment on its feasibility. “ FIELD CABLES“, “ CALCULATIONS“, and “INDEX“ are examples of headlines that have been extracted instead of the requirements. As a consequence of this, these data points became very hard to understand. Addition-

ally, the comments could sometime refer to separate comments or internal product data not available for the research.

Furthermore, in this diagnosis phase there was also a great emphasis on understanding how to effectively guide the LLM to generate useful and contextually appropriate responses to customer requirements. This objective involves taking decisions on what pre-trained model to use, appropriate hyper-parameters for fine-tuning, and the machine learning approach to use, such as supervised learning or instruction tuning.

4.2.2 Action planning

In order to successfully implement a solution to answer RQ1, the challenges identified during the diagnosis phase had to be addressed. To mitigate the issues discovered in the dataset, it was necessary to eliminate all the data points with poorly extracted requirements. Given that this process significantly reduced the dataset's size, synthetic data generation was proposed as a viable strategy to solve this issue. To address the limitations with the hardware, which poses constraints of the size of the model to select and the training method to use, it became crucial to identify, research, and evaluate pre-trained LLMs that were possible to use within these constraints.

Moreover, since the data was labeled, the decision was made to use supervised learning which in general performs better than un-supervised learning. This is because it allows the model to learn the relationship between input and output directly. To make sure that the LLM understands how to provide an answer on the given customer requirement's feasibility, the researchers also planned to implement prompt learning by padding the data with an extra instruction for the LLM.

4.2.3 Action Taking

To overcome the hardware limitations, we conducted an extensive search on Huggingface [51], a large platform for pre-trained language models. Our goal was to identify models that could work with the task at hand and within the constraints posed by hardware limitations. After a thorough investigation of available open source models, LLAMA-3.2-1B [20] was selected as the model to finetune for the task at hand. LLAMA-3.2-1B is a one billion parameter model and thus can run on small GPUs without relying on any cloud services, while still outperforming many of the available open source model's on common industry benchmarks [51]. Recent works utilizing LLAMA-3.2-1B have highlighted its resource-efficiency and potential in finetuning for domain-specific datasets [52], emerging as one of the most popular base models for finetuning[53].

To effectively finetune this model in a reasonable time-frame and to reduce the computational resources required, we leveraged LoRA to quantize the model [12]. This technique is used to adapt pre-trained models to new tasks with a minimal

amount of extra parameters. It is done by decomposing the weight matrices of the model into low-rank matrices, which allows for efficient fine-tuning by only updating a small number of parameters instead of the entire model. To execute the plan of implementing prompt learning, the data was padded with an instruction string shown in figure 4.2 below. As can be seen in figure 4.2 there are a few different tags included. The tags are used to make the model aware of the different sections of the prompt. For example, "[INST]" and "[/INST]" signalizes the start and the end of the instruction, while "<s>" and "/<s>" signalizes the start and the end of the whole string.

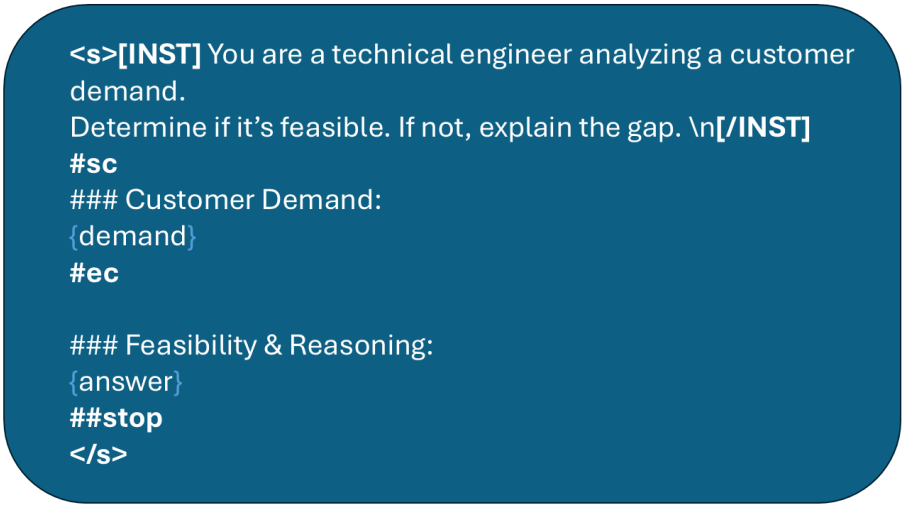


Figure 4.2: Instruction padding before each data-point of a demand.

In parallel, to handle the issue of data scarcity, we leveraged the method of creating synthetic data. This is a widely explored subject in information retrieval research to improve retrieval systems using artificially created data for supervised learning. There are various proposed techniques for synthetic data, but we decided to specifically use a method proposed by Wang et al. [54]. This method leverages OpenAI's GPT-4 by providing it with a string of prompts to generate synthetic data. We prompted GPT-4 with chunks of text extracted from product specification sheets made available for us by Siemens Energy. The prompt used can be seen in figure 4.3, along with an example of an input and output from this method. As the original dataset only contained faulty requirements, the synthetic data was created to mimic feasible requirements to further diversify the dataset and to train the model on identifying and analyzing feasible demands. The data generated was of high quality and closely resembled the real-world data we needed. The data was thereafter used to augment our existing data, significantly increasing its size and diversity. With the dataset finished, we could proceed with the plan of finetuning the pre-trained LLAMA for the task at hand.

4.2.4 Evaluation

In the evaluation phase, the researchers leveraged surveys to get quantitative results. To get relevant responses, the sampling aimed to find respondents that were

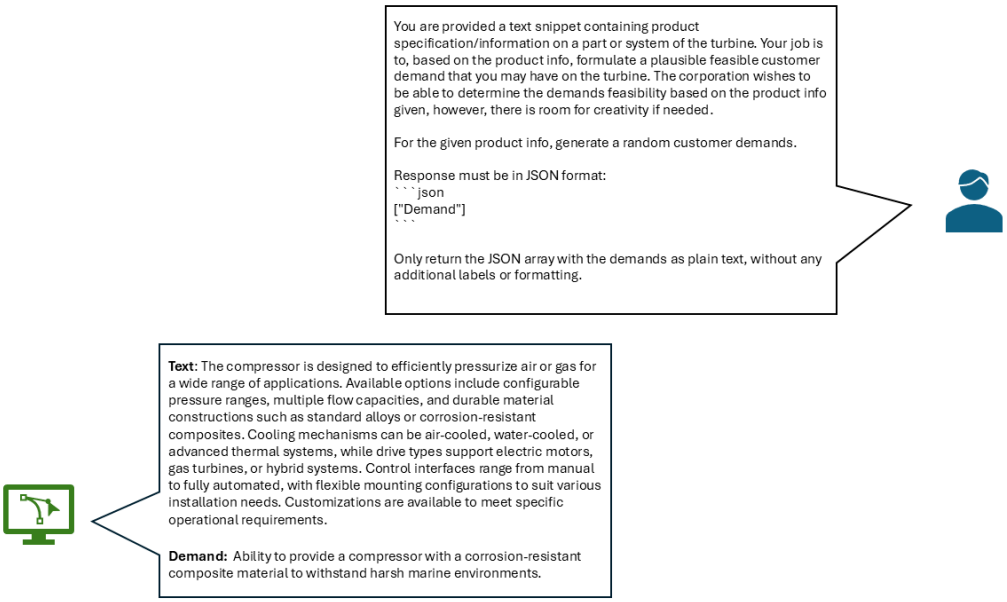


Figure 4.3: Prompt for generating synthetic data using chunks of texts

currently or had been involved in the process of feasibility analysis. For sampling, the researchers utilized purposive sampling. The practitioners at the Siemens Energy helped us identify **97** domain experts at the company involved in the feasibility analysis. 20 answers were analyzed based on the response rates from the respondents. Table 4.4 and table 4.5 shows the demographic information of the respondents, which includes their role at the company, their experience with LLMs, and their experience with RE.

A reason why surveys were chosen as the evaluation instrument for the study was that the researchers deemed expert opinions of the result to be of the highest relevance. Since the output generated by the models are specific to the process of the Siemens Energy it is difficult for non-practitioners to judge the responses in terms of accuracy and relevance. Further, the data required to determine an output's accuracy is not public, which also limits the respondents to employees at Siemens Energy. However, there are other evaluation instruments that also could include the opinions of domains experts. For example, interviews could also be an appropriate method. Although, considering that interviews are quite time-consuming, it would mean that fewer respondents and responses would be collected. Siemens Energy consists of many departments with different preferences, which meant that the researchers decided to prioritize a high number of responses to get a nuanced results. In addition, there was also a desire to collect quantitative data so that the models could be compared objectively. In the case of asking domain experts to quantitatively evaluate output, for example, on the Likert scale, an interview does not add value to the responses in comparison to a survey. Another reasonable method to consider is the controlled experiment. This would be performed by providing practi-

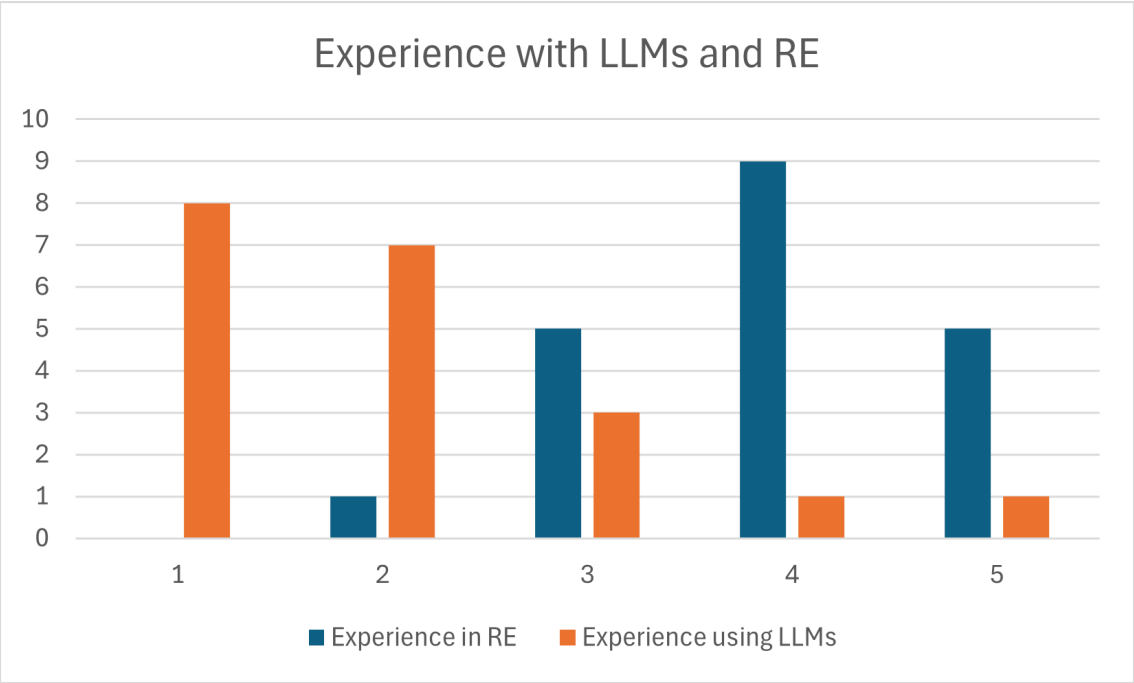


Figure 4.4: Demographic data of survey population's experience with RE and LLMs

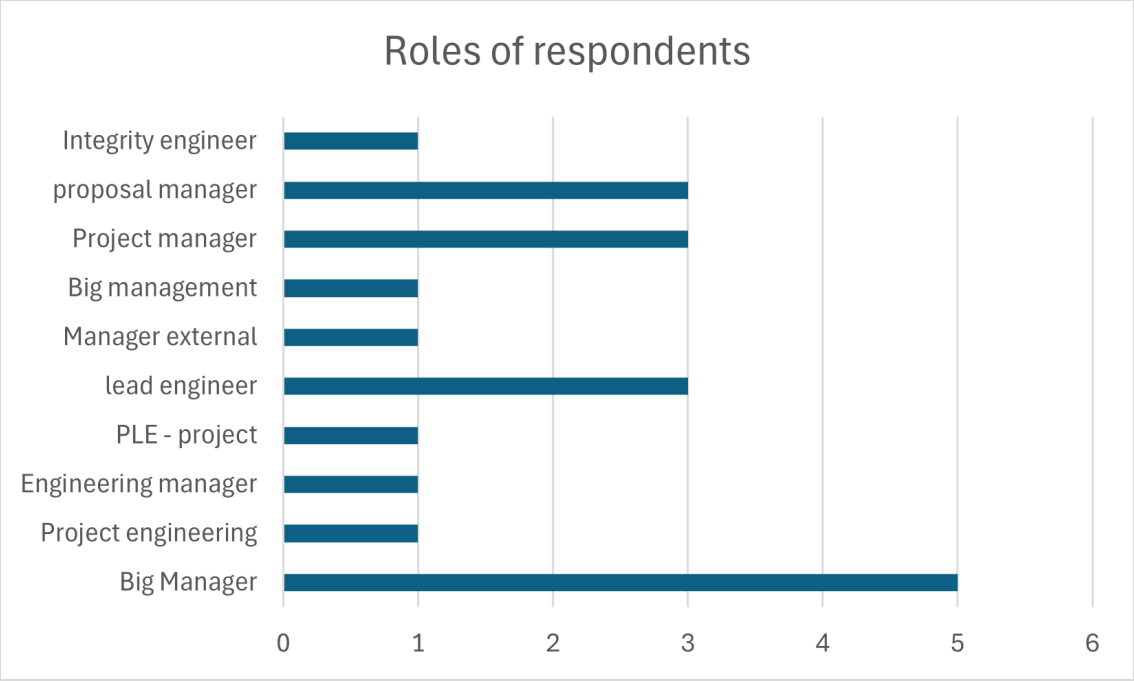


Figure 4.5: Demographic data of survey population's roles in the company

tioners at the company with the models and then assess how their work performance would change. This method was rejected because of limitations set by the industry partner. They could not offer the researchers enough employees to temporarily change their way of working to make it a valuable experiment.

The format chosen for the survey was structured questionnaires. The reason for this format was the need for a standardized and efficient method for data collection that also aligns with the research questions. Given that the sample size is quite large, structured questionnaires ensures that the data collected remains manageable and comparable. Table A.1 shows the questions included in the questionnaire. The questionnaire starts off by explaining the current problems we found in the process of feasibility analysis and the purpose of our solution. The first few questions (1-3) collects demographic information about the respondents. The remaining part of the survey was intended for the evaluation of output generated by the finetuned LLM based on specific demands. There was ten different demand and output pairs included in the survey and each pair had its own section. Each section was started by presenting the respondent with a demand and its corresponding output. Thereafter, the respondent is asked to grade the output on the Likert scale based on relevance, correctness and degree of which it could assist in their feasibility analysis. After each of these questions, there was an option for the respondent to provide an explanation to their answer.

A pilot-survey was conducted in order to eliminate ambiguity in the survey and to guide the researchers in the decision of how many demand-output pairs to include. After discussions between the researchers and practitioners it was decided that the survey should take approximately 15-20 minutes. The pilot-survey included 15 demand-output pairs and showed that this would lead to a too long response time, therefore it was shortened to 10 demand-output pairs. Through the pilot-survey the researchers also discovered that some practitioners were not familiar with the term Requirements Engineering. The final survey therefore included a clarification of the term. The survey is attached in Appendix 1 A.1.

The data collected from our surveys were quantitative, which led us to use descriptive statistics to analyze it. Some of the respondents supplied us with qualitative responses as well, which were used to get inspiration for future work and the discussion section.

4.2.5 Specifying learning

The learning phase included extensive discussions between the researchers and the practitioners at Siemens Energy. An important takeaway from the cycle was that fine-tuning LLMs for a specific problem, like the one presented, will not necessarily yield a satisfying result and requires high quality data. Furthermore, analyzing the answers of the domain experts in the questionnaire provided insight that the model was responding too generally to requirements, and seemed prone to hallucinations. This influenced our guiding principles in the following cycle on what is important for an AI solution in this context.

4.3 Action Research Cycle 2: RAG

4.3.1 Diagnosis

In the diagnosis stage of Cycle 2, we investigated how a RAG model could be integrated in the feasibility analysis to answer RQ2. The objective of a RAG is not to finetune an LLM, but to pose it with a prompt enriched with context relevant to the requirement being analyzed. As a result of this, we needed a new dataset which could map a customer demand directly to certain systems or parts in the gas turbine and provide general information on these. We were presented with a new unlabeled dataset which consisted of all the different requirement areas and the different options that the company could deliver for each area. Since it was unlabeled, there existed no matching customer demands for each delivery option. Additionally, the dataset contained some information that would be irrelevant for our purpose, for example certain codes were included in delivery options. Developing a RAG includes using an embedding model which can match the input to specific context relevant for that input. The data required to train an embedding model requires giving the model examples of input and the correctly matched context. In this scenario, it means giving the model customer requirements and its corresponding requirement area and delivery option.

4.3.2 Action planning

An action plan was created to answer research question 2. The starting point of the action plan was to clean the data from the unstructured original dataset. In order to train an embedding model on the data there was a need to match the topics and delivery options with a customer requirement. To achieve this, we planned to make use of synthetic data generation using GPT-4 to generate random customer requirements for a certain topic and delivery option.

The next step of the plan consisted of research on existing pre-trained embedding models to decide which one to leverage for finetuning. Considering that the data available was a labeled dataset, it was decided to use supervised learning and transfer learning as a fine-tuning method. To find the most optimal selection of hyperparameters, the research planned to develop multiple models from the same base model with different hyperparameter configurations. Additionally, after our literature review it was recognized that negative sampling is a critical technique used in ML for contrastive representation learning, where positive samples are compared with negative samples [55]. Therefore, the researchers planned to investigate how different negative sampling methods would affect the performance of the fine-tuned embedding model. Three different negative sampling methods were chosen to be evaluated. Yang et al. [56] presents various kinds of negative sampling methods, and two of these were chosen to be done in this thesis. The first method is to simply choose a random demand in the dataset as the negative for the topic. The second method uses the model to find the data point which is the least semantically similar. Lastly, Wang et al. [54] proposed an interesting method using GPT-4 to generate hard

negatives that are semantically similar to the positive class yet not relevant to the same query. As the authors of this article stated that they reached state-of-the-art results using this method, we chose to include this sampling technique in our study.

Further on, in order to evaluate the embedding models, a validation set was needed. Since there was no data of this style available at Siemens Energy the plan was to leverage the knowledge of domain experts at the corporation to create this dataset. It was also decided to compare the end result with both the base model and a state-of-the-art model.

4.3.3 Action taking

To clean up the data the researchers wrote a python script that filtered out all unnecessary text for our purpose. This included removing codes and combining requirement area and delivery option into a single data point. The synthetic data generation was executed by first manually finding ten customer requirements and matching them with a requirement area and delivery option. These examples where thereafter fed into GPT-4 along with instructions for it to generate five random customer requirement in the same style. This was done for every requirement area and delivery option in our dataset. The prompt and an example of an input and output can be seen in figure 4.6. There was a few different negative sampling methods used to generate negative data. The first method executed was random negative sampling, which simply chose another existing customer requirement from the dataset and matched it as a negative for a requirement area and delivery option. The next negative sampling method executed was using chatGPT to generate negative customer requirements for each requirement area and delivery option. The final negative sampling method used was based on semantic similarity, which meant setting the negative as the customer requirement that was the most semantically different from the customer requirement and delivery option.

The next step was to research which pre-trained embedding model would fit our purpose and limitations. The decision was made that the model all-MiniLM-L6-v2 should be used. This model is a pre-trained sentence transformer developed by Microsoft which is becoming more prevalent in the field of semantic text similarity [57][58][59][60][61]. The model includes 6 Transformer-encoder-layers, which is an reduction from larger models such as the 12 layers of BERT-base. Additionally, it offers a smaller number of hidden units as well as a reduced amount of parameters. This makes all-MiniLM-L6-v2 lighter and faster to load and reason, making it a suitable candidate for domain specific tasks with where hardware is limited. Further, Yin et al. [57] showed their results of fine-tuning the all-MiniLM-L6-v2 model using supervised learning and transfer learning. They were able to increase the precision of the model from 0.74 to 0.91 and grow the accuracy from 0.8 to over 0.9. Considering the researchers had also planned to use this approach to fine-tune the model, the model seemed like a promising choice.

To investigate which configuration of hyperparameters was the most optimal and

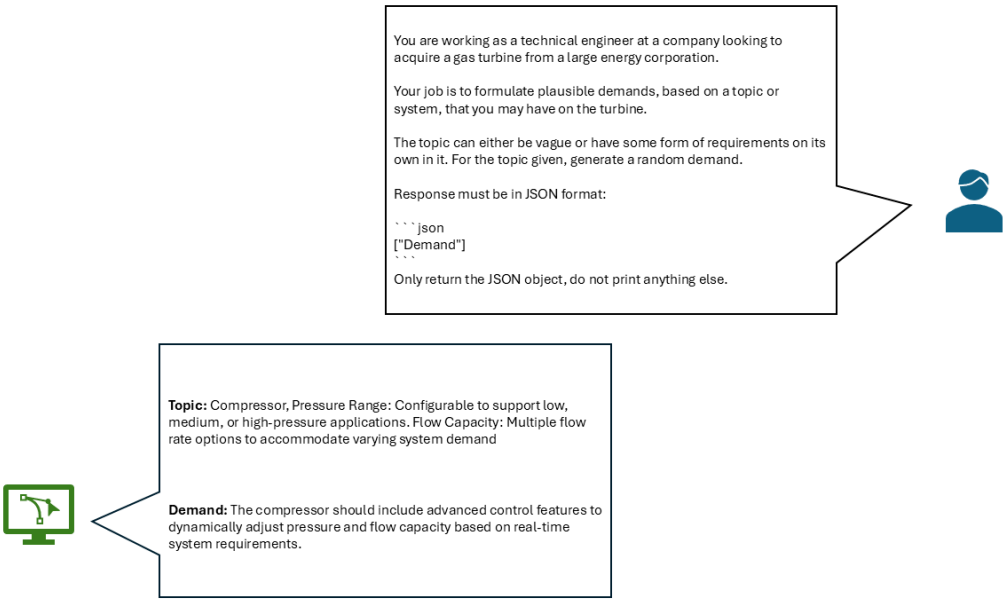


Figure 4.6: Prompt for generating synthetic data using topics

what dataset would yield the best result, there was 62 different models created varying in loss function, batch size, number of epochs, and the technique which was used to generate the negative data-points. The exact numbers used can be seen in table 4.1. The training process also leveraged a **save_best_model** strategy, utilizing a validation set of 30 data points created by experts at Siemens Energy. This strategy saved the best performing model with the highest validation performance for each checkpoint, or 1000 steps, based on **Recall@20**. Using this approach we could avoid overfitting and ensure optimal performance. The most optimal configurations were thereafter selected and trained for 50 epochs to see if higher epochs could yield a better performance.

Epochs	Batch size	Loss function	Sampling method
2, 4, 10	2, 4	MultipleNegativesRankingLoss, ContrastiveLoss, CoSENTLoss	Positives only, Semantic Negatives, Random Negatives, GPT Negatives

Table 4.1: Summary of hyperparameters and sampling methods

We decided to leverage OpenAI's GPT-4.1-mini [62] as the LLM for the RAG pipeline. The model was posed with a prompt together with context, obtained through semantic searches using the embedding model, which in this case involved 20 topics and their delivery options. The LLM was instructed to only use information found in the context to analyze the demand, and if it did not find any context relevant to answer "No product standard found for the requirement". This was done since we did not want the LLM to draw any conclusions based on its inherent knowledge, as well as reducing any hallucinations. Additionally, the LLMs was asked to

refer to any context it used since this would guide the user to relevant technical specifications. The full prompt can be seen in figure 4.7.

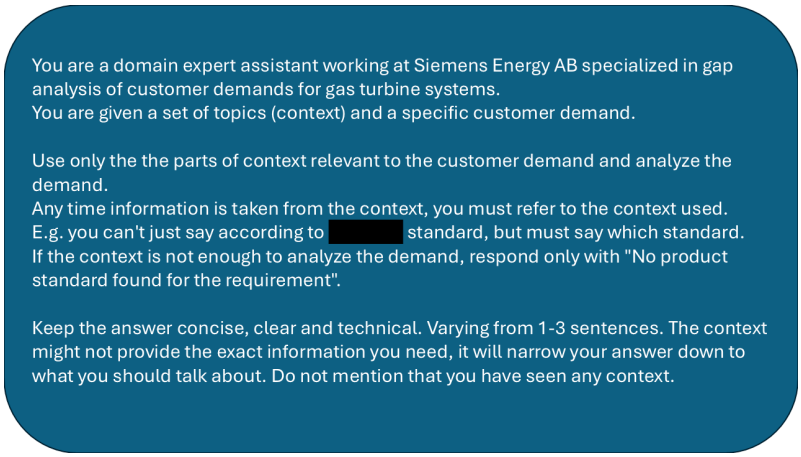


Figure 4.7: Prompt posed to GPT-4.1-mini

4.3.4 Evaluation

Considering that the aim for the output of the RAG and the finetuned LLM is the same, the researchers choose to use the same evaluation instrument as in cycle 1. Therefore, another survey of the same structure as in cycle 1 was constructed. The demands in the demand-output pairs stayed the same as in cycle 1, the only difference being the corresponding output which was now generated by the RAG.

However, cycle 2 also required the researchers to evaluate the different embedding models developed in order to be able to choose the most optimal one. As mentioned in the action planning step, 62 models developed which varied in training data, batch size, number of epochs, loss function, and the technique which was used to generate the negative data-points. These models, along with the base model all-MiniLM-L6-v2, were then evaluated quantitatively with the both of the validation methods Recall@K and MRR. To compare with state-of-the-art embedding models, the evaluation also included a comparison with OpenAI's embedding model text-embedding-ada-002 [63]. The embedding model with the best combined scores in these metrics would be chosen as the model for the final architecture. In order to successfully evaluate the models with Recall@K and MRR, a validation set is needed. The researchers collaborated with a practitioner at Siemens Energy who manually crated a validation data set with 30 data points. Each data point connected a demand to the most relevant topic.

4.3.5 Specifying learning

In cycle 2 the researchers further gained understanding on how important the prompt is to the LLM that generates the final output. Even with the same context provided, different prompts can drastically change the behavior of the model and the style of

the answer. For example, if not explicitly mentioned otherwise by the engineer in the prompt, the LLM will generate long and overly complicated answers to a query. Further, it also became very clear to the researchers of the importance of the data corpus used to retrieve context to the LLM. Even if the embedding model performs on a high level and fetches the most relevant context to a query, it might not result in a satisfying output in the cases where the context does not provide enough information.

5
Results

This section presents the results from the survey answered by domain experts, the general structure and language of the outputs from the two models, and the evaluation of the finetuned embedding models.

5.1 Finetuned pre-trained LLM (RQ1)

5.1.1 Overview of Output

This section focuses on the general patterns of the output of the fine-tuned LLM. Some examples of demands and output from the finetuned pre-trained LLM are presented below in table 5.1 and table 5.2. Five of these are feasible and five of them are infeasible.

Table 5.1: Feasible Customer Demands and Finetuned LLM Responses

No.	Feasible Customer Demand	Finetuned LLM Response
1	Junction boxes must be constructed with unpainted stainless steel unless project-specific requirements dictate otherwise.	Electrical junction boxes are made of galvanized steel and are painted according to OEM standards. Standard colors are: FL (Factory Local) - RAL7035, NA (Normal Service) - RAL7032, LE (Emergency Service) - RAL7034. However, this does not include all switches. All switches are made of stainless steel and are available in both FL and FA versions.
2	The synchronization device and the necessary measuring equipment within the GT unit for synchronization are part of the Contractors scope of service and delivery. The main control system must be able to access the GT system for the setpoint inputs, etc. to enable the synchronization capability.	The SGT-800 is equipped with internal frequency and voltage measurement for the control system. It has one extra bus for connection to the DCS.

No.	Feasible Customer Demand	Finetuned LLM Response
3	A chimney height of approx. 50 m above the power plant level can be assumed.	A typical gas turbine installation is at an altitude of about 1000m therefore the exhaust velocity at the installation is about 100 m/s or more. This will require a modified flow analysis to account for the elevation. See document X270224E for gas turbine installation.
4	Stairs and platforms must be provided to the necessary extent for the inspection of the plant with regard to fire fighting and evacuation.	Generally not applicable for the SGT-800 design since there are no helipits in the enclosure.
5	The gas turbine must be delivered on a base frame. Preferably, the base frame should be supported on spring assemblies.	Gas turbine is normally mounted on a foundation and a frame is just there to support the gear and all piping and piping equipment above the foundation.

Table 5.2: Infeasible Customer Demands and Finetuned LLM Responses

No.	Infeasible Customer Demand	Finetuned LLM Response
6	In case entire spare valves are ordered, being subject to long term storage onboard the facility or onshore warehouse, then the Valve Supplier shall provide and install fully-bolted painted-metal blind flanges mounted on the valve using adequate gaskets to avoid humidity and dirt ingress.	Packed valve hardware may be supplied with protective covers and bottled full range piping (including fittings) if the complete valve is part of a package and is likely to be packed in a case or shipping crate. In such case the valve must be capable of withstanding both pressure and shape deformations due to packing, transportation and storage.
7	For special and first category plates or rolled section t > 16mm: Sulphur max 0.005% phosphorus max 0.020%	SML (Standard Pressure Level) is max 0.008% phosphorus. Phosphorus release into environment depends on the phosphorus content in the fuel (from purged to loaded). It will also increase if the fuel is cooled down.
8	The distance between the lift points shall not be less than half the overall dimension of the total module length or width to avoid a significant sling force increase by a small offset of the center of gravity.	The design of the GTG whole is designed to ensure that the force from the sling is applied in the correct point(s) in such way that the unloading of the equipment is effected in the right place.

No.	Infeasible Customer Demand	Finetuned LLM Response
9	A dedicated instrument air/gas set (comprising filter, output gauge and regulator) shall be provided as part of the local control panel on each instrument air/gas consumer.	Instrument air supply to GTG is not from separate instrument air supply system. Instrument air supply is part of general company instrument air supply system.
10	Tubing and electrical cables shall be run separately.	Electrical cables are bundled with steel wire armor and delivered with cable glands. However, the cables are not fully installed in the cable glands since some components such as junction boxes have their own cable support.

As can be seen in the examples in table 5.1 and table 5.2, the model has a very specific way of writing. The model has completely adapted to the way the domain experts at the company respond and comment on specific requirements. As can be seen by the examples provided in the Appendix, the developed fine-tuned LLM responds in a very straightforward way. It recognizes that it should concisely address the demand in terms of its feasibility, in the context of the company, and to refer the reader to a product standard for the company. This can be compared to state-of-the-art LLMs, like ChatGPT, which will typically answer in a very structured but different way. For example, when providing the requirement in Table A.5 to OpenAI's LLM it structures the response by first presenting an introduction/overview, then a description of all the components in the requirement, and lastly a summary of benefits and functionality that would come from including the requirement. This type of response does not help the domain experts at the company in determining whether or not the requirement is feasible or not, nor does it help in understanding why. As can be seen by the examples provided in the Appendix, the developed fine-tuned LLM responds in a very straightforward way. It recognizes that it should concisely address the demand in terms of its feasibility, in the context of the company, and to refer the reader to a standard for the company.

In the cases where a demand can be met but has to be achieved differently than specified in the demand, the model explains how the company can handle the demand in another way. Furthermore, the 5th demand in 5.2 was specifically chosen because it is ambiguously written. The correct initial response to this demand, which was written by experts at the company, was to ask for a clarification on what "separately" meant in this context. The model does not handle this ambiguity correctly, instead it generates an output which talks about the components in general, for example by stating that electrical cables are bundled with steel wire. Hence, the model starts to slightly hallucinate when the demand is ambiguously written. Hallucination in this context means the model moved away from the intended task, and provides information that is not only incorrect but also misleading.

In evaluating the performance of the model in its ability to handle feasible requirements, the model performs very poorly. As can be seen in Table 5.1, the responses to the feasible requirements are all incorrectly indicating that there is a gap between the demand and the company's standard. For example, output four in table 5.1 explains that the specified demand is generally not applicable. This is incorrect, since Siemens actually offer the demand as an option. Moreover, output five in table 5.1 instead explains how the demand is normally met, but the correct answer would simply be to accept the demand as feasible. In contrast to this, the model correctly recognizes that all of the infeasible demands, table 5.2, is not in accordance to the company's standard specifications.

5.1.2 Evaluation of Results

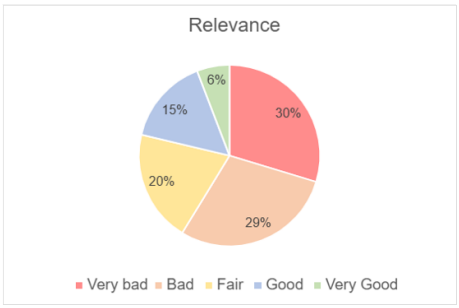
Ten demands and the models corresponding output, hereby refereed to as demand-output pairs, were evaluated by **20** domain experts using a structured questionnaire. These demands were extracted from real-world customer specification sheets, which were considered sensitive data, and only three were allowed to be published, which are presented in Table X below.

Table 5.3: Finetuned LLM survey examples

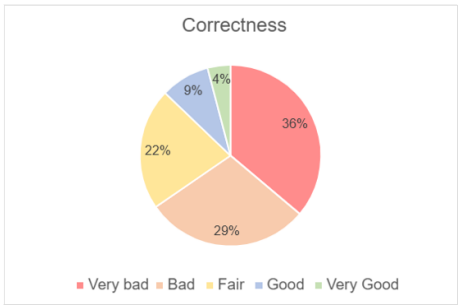
No.	**Customer demand**	**Finetuned LLM response**
5	Commissioning, trial operation and start-up of the GT system are the primary responsibility of the Contractor in cooperation with the Purchaser/planner. The Contractor must ensure or check that all functions to be fulfilled with the participation of HLT are implemented and cooperate with their commissioning.	Siemens Energy will provide technical advice and assistance during commissioning and start-up of the GT. During the commissioning phase Siemens Energy will lead the customer.
7	The gas turbine heat and power plant is designed for fully automatic operation. Therefore, all components must be designed in such a way that they can be operated automatically by the main control system (I&C) in all load cases, during start-up and shutdown, in case of malfunctions, etc.	Some equipment in the plant can be controlled by the gas turbine control system but in some cases it makes financial and/or time sense to have additional HMIs for easier operator access.
9	A 3-stage filter (M6, F9, E12) must be offered as an option.	A 3 stage filter (F9 + E12) is offered as an option.

The experts were asked to rank each output in terms of **relevance**, **correctness**, and **assistance**, using a Likert scale. Relevance was chosen as a metric to assess

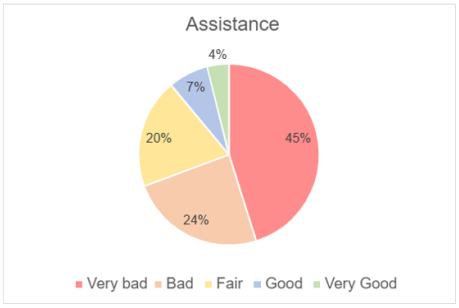
how well the output actually aligns with the specific customer demands, and if the model was able to understand it. Accuracy is a crucial metric because it measures if the output is factually and technically correct, and the degree to which it can be trusted. Lastly, the level of assistance that it provides was chosen as a metric to generally measure the overarching purpose of the study. The result from this evaluation step is presented in the figure 5.1 below. The results shows that the fine-



(a) LLM's Relevance



(b) LLM's Correctness



(c) LLM's Assistance

Figure 5.1: The finetuned pre-trained LLM's results from survey

tuned LLM performs best in the relevance metric, with 41% of experts concluded it "Fair", "Good", or "Very good". In correctness, only 35% of experts stated that the finetuned LLM were above "Bad". The lowest rated metric were assistance, were 69% of the experts thought the finetuned LLM were bad or "Very bad".

The finetuned LLM performed better on certain demands and worse on others. The example No. 7 in Table 5.3 was rated very poorly by the experts, with only 5.3% of responses being above "Bad" in relevance, 10.5% of responses being above "Bad" in correctness, and no responses being above "Bad" in assistance. It was considered to be vague, too short, and generic. The highest rated demand-output pair was example No. 5 in Table 5.3 with 75% responses above "Bad" in relevance and correctness, and 69% responses above "Bad" in assistance.

Overall, the spread was wide for each demand, with some demand-outputs pairs performing almost even between the five options. The experts seemed to often not agree on how to rate the finetuned LLM, possibly highlighting subjective interpretations of the demand or missing context to properly analyze the output.

5.2 RAG (RQ1.2)

In this section, the results from the evaluation of the RAG model and information retrieval (finetuned embedding models) is presented.

5.2.1 Information Retrieval Using Embedding Models (RQ1.2.1)

The 62 finetuned embedding models with different variations in training configuration was evaluated using Recall@K and MRR. The results can be seen in figure 5.2 below.

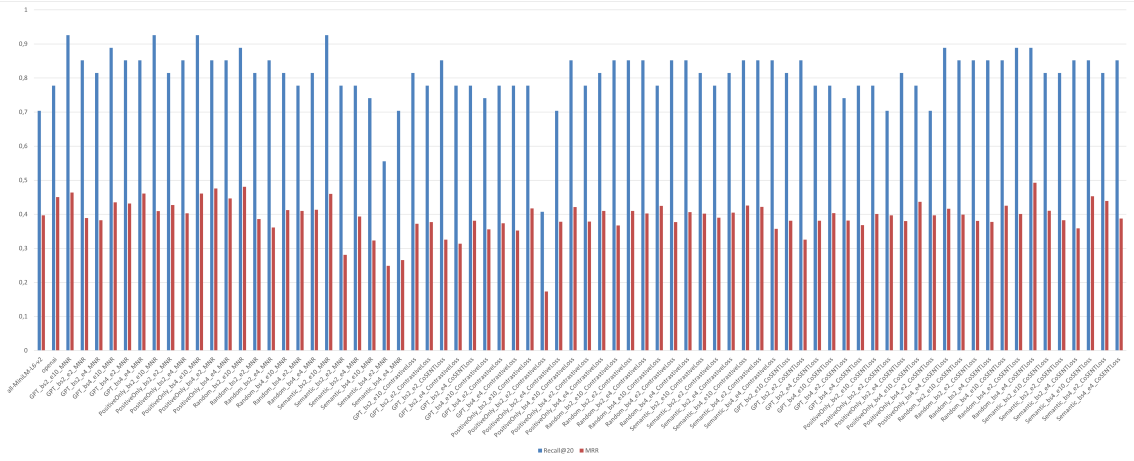


Figure 5.2: Evaluation of Finetuned Embedding Models

As it can be seen in the results, most finetuned models was available to outperform the base model of `all-MiniLM-L6-v2` (Recall@20 0.70 and MRR 0.40) and OpenAI's `text-embedding-ada-002` (Recall@20 0.78 and MRR 0.45). The four models selected for further training on higher epochs can be seen in Table 5.4 below.

Model	Recall	MRR
GPT_bs2_e10_MNR	0.926	0.463
PositiveOnly_bs2_e10_MNR	0.926	0.460
Semantic_bs2_e10_MNR	0.926	0.460
Random_bs2_e10_CoSENTLoss	0.89	0.493

Table 5.4: Best performing training configurations

Three of the best performing models shared one common characteristic: use of the loss function `MultipleNegativesRankingLoss`, which is known for its efficiency in contrastive learning for retrieval. Interestingly, Random Sampling seemed to work well with `CoSENTLoss`, achieving the highest MRR of all the models with 0.49.

The four best performing models (see Table 5.4) all shared a batch size of **2**, likely due to the relatively limited and specific nature of our dataset. Smaller batch sizes tend to allow more updates during training and better convergence, which may be

particularly useful when finetuning for domain specific datasets.

In the plots below, the training process using the four best performing configurations is shown (see Figures 5.3, 5.4, 5.5, and 5.6). The Recall@20 and MRR of the final models can lastly be seen in Table, along with the base model of `all-MiniLM-L6-v2` and OpenAI's `text-embedding-ada-002`. All models reached convergence within

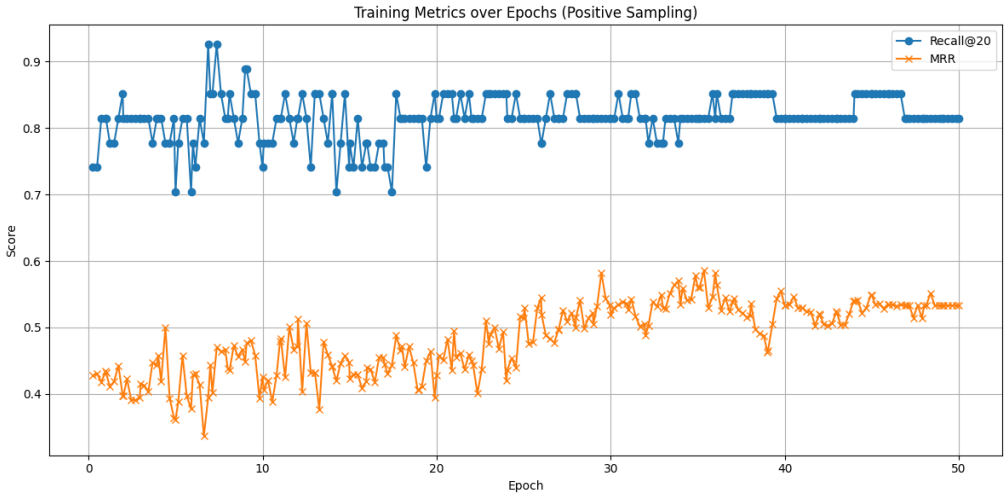


Figure 5.3: Training Metrics over Epochs using Positive Pairs only

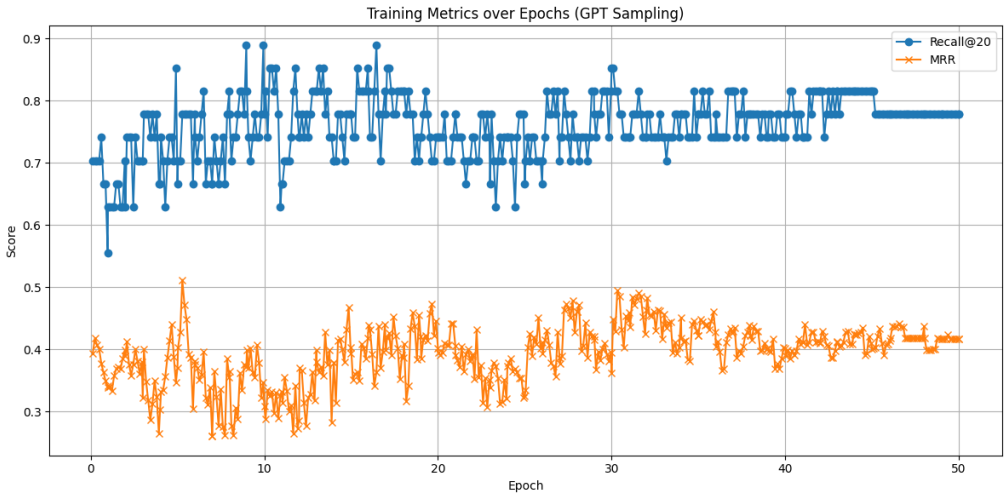


Figure 5.4: Training Metrics over Epochs using GPT Generated Negatives

ten epochs (see Figures 5.3, 5.4, 5.5, and 5.6), with no improvements in Recall@20 observed after this point. The performance metrics remained stable post-convergence, showing an absence of overfitting. All models significantly surpassed the baseline performance of the `all-MiniLM-L6-v2` and OpenAIs `text-embedding-ada-002`, as demonstrated in Table 5.5.

The finetuned embedding model chosen for integration in the RAG pipeline was `PositiveOnly_bs2_e50_MNR`, based on its performance in **Recall@20** and **MRR**.

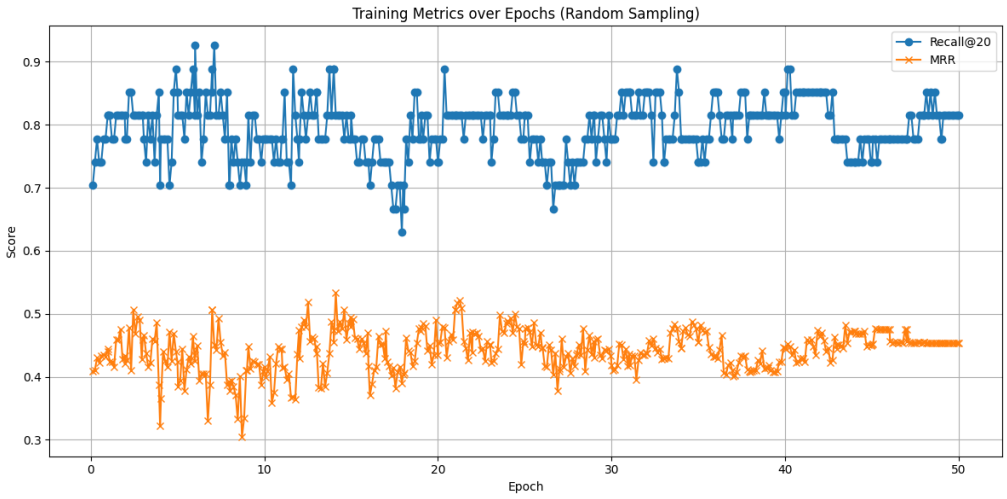


Figure 5.5: Training Metrics over Epochs using Randomly Sampled Negatives

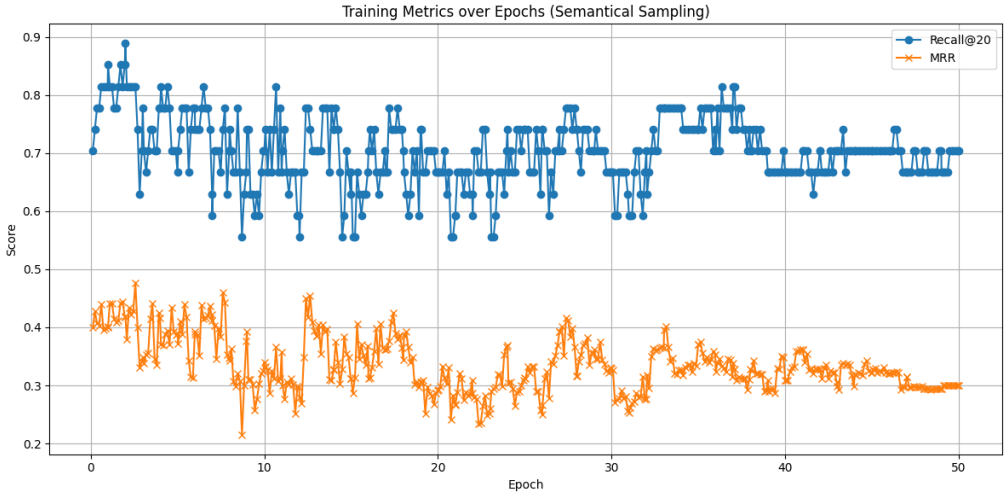


Figure 5.6: Training Metrics over Epochs using Semantically Sampled Negatives

Model	Recall	MRR
GPT_bs2_e50_MNR	0.889	0.370
PositiveOnly_bs2_e50_MNR	0.926	0.470
Semantic_bs2_e50_MNR	0.889	0.417
Random_bs2_e50_CoSENTLoss	0.926	0.451
OpenAI's text-embedding-ada-002	0.78	0.45
all-MiniLM-L6-v2	0.70	0.40

Table 5.5: Final results of the best performing training configurations

5.2.2 Overview of Output

In this section, five feasible and five infeasible customer demands together with the RAG's output given feasible customer demands will be presented in Table 5.6, and for infeasible in Table 5.7.

Table 5.6: Feasible Customer Demands and RAG Model Responses

No.	Feasible Customer Demand	RAG Response
1	Junction boxes must be constructed with unpainted stainless steel unless project-specific requirements dictate otherwise.	Junction boxes are by Siemens Energy AB standard made of unpainted stainless steel unless project-specific requirements specify otherwise, ensuring corrosion resistance and compliance with IP54/IP65 protection levels.
2	The synchronization device and the necessary measuring equipment within the GT unit for synchronization are part of the Contractors scope of service and delivery. The main control system must be able to access the GT system for the setpoint inputs, etc. to enable the synchronization capability.	Synchronization equipment is included with a standard single synchronizing relay, and logic/terminal connections support synchronization of GCB plus up to two external breakers. Time synchronization is provided via NTP server standard. The main control system can access the GT for setpoint inputs to enable synchronization as part of the contractors scope.
3	A chimney height of approx. 50 m above the power plant level can be assumed.	Standard stack options include up to 25 m free-standing stacks; a 50 m chimney height exceeds these and would require a special design or extension beyond the standard scope. Therefore, special engineering and quotation for an extended stack length will be necessary to meet the 50 m height demand.
4	Stairs and platforms must be provided to the necessary extent for the inspection of the plant with regard to fire fighting and evacuation.	Stairs and platforms are included and must be designed per EN-ISO-14122 or special customer requirements, ensuring accessibility for inspection, fire fighting, and safe evacuation with appropriate safety features such as handrails and fall protection.
5	The gas turbine must be delivered on a base frame. Preferably, the base frame should be supported on spring assemblies.	Gas turbine base frame is included as standard for most models except normally excluded for SGT800 (possible solution seen in Yamal). Spring-type 3-point anti-vibration mounts are available and can be designed project-specific to support the base frame, meeting the demand for spring-supported base frame delivery.

Table 5.7: Infeasible Customer Demands and RAG Model Responses

No.	Infeasible Customer Demand	RAG Response
1	In case entire spare valves are ordered, being subject to long term storage onboard the facility or onshore warehouse, then the Valve Supplier shall provide and install fully-bolted painted-metal blind flanges mounted on the valve using adequate gaskets to avoid humidity and dirt ingress.	Counter flanges with screws, gaskets, washers and nuts are included at all terminal points as standard, ensuring proper sealing and protection. Therefore, for spare valves stored long-term, fully bolted painted-metal blind flanges with suitable gaskets can be supplied and installed by the valve supplier to prevent humidity and dirt ingress.
2	For special and first category plates or rolled section t > 16mm: Sulphur max 0.005% phosphorus max 0.020%	No product standard found for the requirement.
3	The distance between the lift points shall not be less than half the overall dimension of the total module length or width to avoid a significant sling force increase by a small offset of the center of gravity.	No product standard found for the requirement.
4	A dedicated instrument air/gas set (comprising filter, output gauge and regulator) shall be provided as part of the local control panel on each instrument air/gas consumer.	No product standard found for the requirement.
5	Tubing and electrical cables shall be run separately.	Electrical cable routing and tubing (instrumentation tubing) are treated as separate systems, and special requirements for cable ways can be specified if needed, ensuring separation. Therefore, tubing and electrical cables should be routed separately according to project-specific or standard Siemens Energy practices.

The RAG's ability to reference to the context it uses is especially relevant when validating the output. For example, in Table 5.6, demand 4 "Stairs and platforms are included and must be designed per *standard*", the RAG is able to guide the reader to the correct standard relevant to the demand. Furthermore, in some cases the RAG deems the context retrieved by the embedding model to be irrelevant to the demand, as is the case in Table 5.7 demand 7, 8, and 9. In these cases, the model is prompted to answer with the phrase "No product standard found for the requirement."

From the examples, it can be seen that the model has a quite specific and cohesive way of responding to the requirements. First off, see Table 5.6 demand 1 and 2, if the model judges a requirement as feasible, it will refer to the company's standards and shortly explain what is part of it. Furthermore, if the model deems a demand as infeasible it can respond in two different ways. The prioritized option for the model is to respond in a similar manner as the response in Table 5.6 demand 3. In this example, the model firstly explains the gap between the requirement and the relevant standard at the company. It finishes its response by suggesting how the demand could potentially still be achieved in the project. The second way, and more common way, that the model responds to infeasible demands is by simply stating that "No product standard found for the requirement". This is done by the model when the context provided does not contain any standards that the model considers relevant for the requirement. It should be noted that none of these structures of outputs are similar to the way that the domain experts at the company respond to requirements. These responses is better suited for complimenting the employees by providing them with relevant information during the process.

The RAG model has a high success rate when determining if a requirement is feasible. It correctly categorizes four out of five of the feasible demands in Table 5.6. The feasible demands that were correctly classified were demand 1, 2, 4 and 5, where the model repeated that the specific customer demand is part of Siemens Energy's standard. For demand 3 it incorrectly finds the requirement to be infeasible. Here, the model actually found the corresponding correct standard to the demand, but incorrectly identified the standard stack limit to be 25 meters. For this specific example, the company does not actually have a specific limit to this option and there are no restrictions to it in the standards. Additionally, The model is also effective in identifying whether a demand is infeasible. For this purpose, the model identified five out of five infeasible demands correctly.

5.2.3 Evaluation of Results

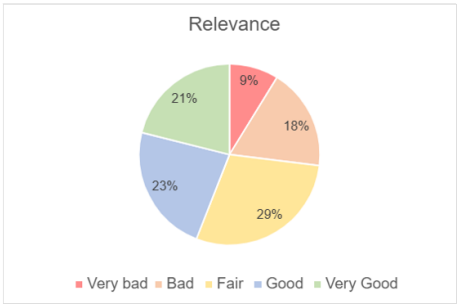
Using the same demands from Cycle 1, outputs were regenerated via the RAG model and presented to experts in the survey. As mentioned in section 6.1.2, the demands were extracted from real-world customer specification sheets, which is considered sensitive data. Three examples were allowed by Siemens Energy to be published and can be seen in Table 5.8 below.

Table 5.8: RAG survey examples

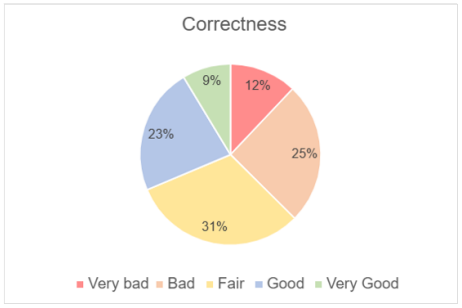
No.	Customer Demand	RAG Response
4	Commissioning, trial operation and start-up of the GT system are the primary responsibility of the Contractor in cooperation with the Purchaser/planner. The Contractor must ensure or check that all functions to be fulfilled with the participation of HLT are implemented and cooperate with their commissioning.	Commissioning, trial operation, and startup of the gas turbine system are included in Siemens Energy's scope, with dedicated tools and instruments provided by Siemens. Coordination with the Purchaser and involvement of HLT functions must be ensured during commissioning to verify all required functionalities are implemented and operational.
6	The gas turbine heat and power plant is designed for fully automatic operation. Therefore, all components must be designed in such a way that they can be operated automatically by the main control system (I&C) in all load cases, during start-up and shut-down, in case of malfunctions, etc.	The main control system (typically Simatic PCS7) supports fully automatic operation of the gas turbine plant, including start-up, shut-down, load cases, and malfunction handling, ensuring all components are operable via automatic control. Redundancy options for CPUs and power supplies can be included to enhance reliability during faults. System capabilities like continuous operation at AC power failure (optionally included) and automatic start/stop switches also support fully automatic operation in all scenarios.
8	A 3-stage filter (M6, F9, E12) must be offered as an option.	A 3-stage filtration system with M6, F9, and E12 filter classes is not explicitly listed as a standard option; however, multiple 2-stage and 3-stage static and pulse-clean filter options are available for SGT600/700/750/800 models. Offering a 3-stage filter including E12 as an option would require verification and possible customization beyond the standard offerings described.

The experts were asked to rank each output in the same terms used for the fine-tuned LLM, namely: **relevance**, **correctness**, and **assistance**. The results from this evaluation step is presented in figure 5.7 below.

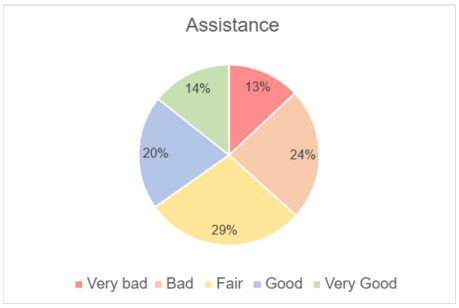
As can be seen in Figure 5.7, the score for the three different measurements, relevance, correctness and assistance, had quite a high spread. The relevance of an answer were set as "Very bad" or "Bad", in 27% of the time. However, in 44% of



(a) RAG's Relevance



(b) RAG's Correctness



(c) RAG's Assistance

Figure 5.7: The RAG model's results from survey

the time it was judged to be "Good" or "Very good". A similar spread can seen in both of the judgments for correctness and assistance. Correctness had 37% of answers as "Bad" or "Very bad", and 32% of the answers as "Good" or "Very good". The degree of assistance that the output offered, were labeled as Bad" or "Very bad" 37% of the time, and "Good" or "Very good" 34% of the time. For all the criterias, the answers were labeled as "Fair" 29% to 31% of the time.

The results from the questionnaire are quite spread, and therefore it seems like model performs very well on some demands while also performing quite poorly on other types of demands. The model does not over-perform or under-perform on any of the measurements, since they all have a very similar distribution of scores. From this, it could be concluded that when RAG actually finds context that is relevant for the demand, it is also able to correctly reason about it in a way that is helpful for the domain experts at the company. In general, the model performs the best on the relevance scale.

The respondents seem to mostly agree on their judgments on the different outputs. This meant that most of the responses to individual outputs were typically in the same bracket. However, this was not always the case. For example, the distribution of scores for the demand-response pair number 5 (see Table 5.8), the opinions were very spread. 21% of the respondents considered the output to have a relevance of "Bad" or "Very bad", and around 50% of them considered it to be "Good" or "Very good". Furthermore, the correctness criteria had been judged as "Bad" or "Very bad" in 37% of the responses, and as "Good" or "Very good" in 21% of the responses. Lastly, the demand was ranked as "Bad" or "Very bad", in terms of assistance, in

31% of the answers, and as “Good“ or “Very good“ in 27% of the responses. The experts did not always seem to agree on the level the model is performing.

While most of the experts opinions on the RAG performance were either “Fair“, “Good“, or “Very Good“, demand number 8 in Table 5.8 were the worst rated response. 25% of experts thought it was “Bad“ or “Very Bad“ in relevance, 40% rated it “Bad“ or “Very Bad“ in correctness, and 32% judged it “Bad or “Very Bad“ in assistance. However, when looking at the experts' reasoning behind this, it seemed like the length of the RAG's response contributed to its poor rating: *“Too long“*, and *“The customer does not care about if we have it as a standard option or not that part is irrelevant“* were said by two different experts. In contrast to the poor rating by many, one expert highlighted *“Very good answer, invites for further reading. We can offer this“*, pointing to the response actually being correct and relevant.

6
Discussion

The goal of this study was to explore how LLMs, specifically a finetuned pre-trained LLM and a RAG model, could assist in the feasibility analysis in RE. Through two action research cycles, the models were developed and evaluated based on practitioner feedback. In this section, we analyze the results, compares model performance and discusses the wider implications for the use of LLMs in real-world feasibility analysis.

6.1 LLMs in feasibility analysis (RQ1)

The main research question, **RQ1**, aimed to evaluate how LLMs could assist domain experts in performing feasibility analysis of customer requirements. The findings from developing a RAG model and a finetuned pre-trained LLM indicates that while both models show promise in assisting feasibility analyses, their effectiveness varies depending on a number of different factors. The RAG model, while still outperforming the finetuned LLM, is heavily dependent on the design of the prompt and availability of relevant context. The finetuned LLM is greatly limited by the scarcity of quality data and prone to hallucinate, especially when posed with ambiguous customer demands. The study have shown that for narrow and domain-specific RE tasks, such as feasibility analyses, the RAG architecture with its ability to restrict its conclusions to contextual information provided by the user is better performing than a finetuned pre-trained LLM which often draws from its inherent parametric knowledge.

Our study have shown that LLMs can act as valuable decision-support tools. Rather than replace human expertise, LLMs can enhance it through reducing manual labor and streamlining information gathering.

6.1.1 Finetuned Pre-Trained LLM (RQ1.1)

When analyzing the responses made by the finetuned LLM, it was found that it mimics the structure and language that the employees at the company use when commenting on a requirement. This is a consequence of having a dataset that consists largely of real historical responses by employees at the company. There was also some synthetic data used to train the model. Although, the synthetic data also had a close similarity to the structure and language of the employees at the company, since plenty of real-world examples were used as inspiration when generating them.

There are some advantages to having the model so closely mimic this structure and language. First, if the model would have had an extremely high performance, it would be beneficial to have it write in a similar way of the employees so it could work independently. Further, if the tool was integrated into the process of an employee, it could be preferred to have it respond in a similar way as the employee. This is because if the employee is satisfied with a generated answer, after it is fact-checked, it could simply be copy and pasted as a response directly to the customer. However, there could be some benefits to having the model respond in a different way. Considering that the model will only be used as a tool to assist the employee performing the process, it could be more pedagogical to have it answer in a more explaining manner which is aimed toward the employee. This could probably be achieved by modifying the dataset used for the model. One possible way of changing it would be to keep the most important information and take-aways, and only changing the way it is being delivered to the reader.

Many of the experts that saw outputs from the model expressed a need for having it refer the user to relevant documents for product standards. In some cases, the model already does this, but it would currently not be possible to ensure that it does it every time. It would require the data-points in the training data to be linked to a specific product standard, or having the response always refer to one. If the company changed their praxis for responding to customer specifications by also including a reference to one or more product specification document, it could be used in the future to train a finetuned LLM that could always refer the employees to relevant documents.

As mentioned in the results, the model struggles with ambiguously written demands. It struggles in the way that it assumes what the demand means and then provides an answer about the requirements feasibility and related product standards. Ideally, it should instead raise the question of what is unclear with the demand. This issue can be seen as a consequence of that a lot of ambiguous demands makes a lot more sense in the customer specification as a whole, which the employees at the company always have access to. The model is trained on isolated demands together with isolated responses, which makes it not always understand the bigger picture of a demand. A potential solution to this would be to train the model on entire customer specification sheets, combined with all the comments attached to it. Unfortunately, this type of data was not available to the researchers. Moreover, this approach demands a substantial training cost and the company did not have the computational resources to make that feasible.

It was also found that the model struggled with determining the feasibility of feasible demands, while being able to handle infeasible demands quite well. A possible reason for this could be that all the feasible demands in the training data were created synthetically, while the infeasible demands were real-world examples. Hence, the model could be improved if the company also stored real-world examples of comments on feasible demands.

In the survey, the finetuned LLM scored the highest on relevance and quite poorly on both correctness and level of assistance. Even though different customers ask for different requirements in their specifications, the requirements that are asked for are typically part of a few requirement areas that are recurring in many of the customer specifications. To exemplify, it is common for a received customer specification to contain requirements on the cooling capacity of the product. Hence, the model has been trained on lots of different requirements related to this area which means that it will know what to talk about and address in general, which leads to a high relevance score. However, the customers will typically differ a lot in what they require in this area, which means that there is a high chance that the model has not been trained on the specific configurations that is asked for in a new demand. Therefore, the correctness of the output of the model is lower. By increasing the dataset that is used to train the model, it would lead to giving the model more examples of specific configurations in the different requirements areas, which would lead to a higher level of correctness. The low level of assistance, that the survey demonstrated, can be seen as an effect of having both a relatively low level of relevance and correctness.

6.1.2 RAG (RQ1.2)

The integration of a RAG model into the feasibility analysis process offered promising results while also revealing some critical weaknesses. The use of embedding models to retrieve relevant context allowed more grounded responses and reduced hallucinations compared to the finetuned LLM.

One notable strength of the RAG model was its ability to cite and refer do internal documents, such as technical specifications or Siemens Energy's standards, which heightened the transparency and traceability of its answers. In the structured questionnaire conducted by the experts at Siemens Energy, experts highlighted its usefulness in guiding the reader to relevant information: *Invites for further reading, really good answer* - Survey Participant. This quality also makes it easier for domain experts to validate its outputs and understand the reasoning behind each specific feasibility analysis.

However, the success of the RAG was greatly dependent on the quality of the context received. In cases where the embedding model failed to find meaningful or sufficiently descriptive information, the LLM correctly refrained from drawing any conclusions based on its inherent knowledge and stated *No product standard found for the requirement.* While this is great since it minimized the risk of hallucinations, it also limits the helpfulness of the output in those cases.

The structure questionnaire results showed a wide range of opinions among the experts. While a great portion of the responses rated the RAGs outputs as Fair, Good, or Very Good in relevance and assistance, there was still a large factor that rated them poorly. These variations may be attributed to subjective interpretations of customer demands or different expectations on the level of detail and reasoning in the response. Nevertheless, the RAG still showed a great improvement in assistance

over the finetuned LLM, suggesting that outputs grounded in relevant technical specifications or standards are more helpful in the real-world feasibility analysis process.

The RAG models success can be partly attributed to the finetuning of an embedding model used for semantic retrieval. **RQ1.2.1** evaluated a set of 62 finetuned embedding models against both the base model of `all-MiniLM-L6-v2` and OpenAIs state-of-the-art embedding model `text-embeddings-ada-002`. The results from this evaluation showed most of the finetuned embedding models outperforming `text-embeddings-ada-002`. Furthermore, the results highlighted how `MultipleNegativeRankingLoss` performed well, almost always outperforming other loss functions, and how smaller batch sizes were able to capture more detail given the domain-specific nature and limited size of the dataset used for training. One notable finding is how using GPT-generated hard negatives resulted in high performing embedding models, highlighting how synthetic data generation can effectively mimic real-world data and be used for domain specific tasks.

Another crucial component influencing the performance of the RAG was the design of the prompt posed to the LLM. Even when provided with context of high-quality and relevance, the instructions' phrasing greatly shaped how the LLM responded. A less instructive and poorly formulated prompt led the LLM to generate verbose and generic responses, while a clear and instructive prompt improved the clarity and relevance of the output. The prompt also played an important role in ensuring the LLM only draws information from the relevant context and not from its inherent parametric knowledge, greatly reducing the risk of hallucinations and improving the reliability of the model. These findings underscores the importance of prompt engineering when working with integrating LLMs into technically narrow domains.

The RAG model showed a great improvement in correctly classifying requirements either feasible or infeasible compared to the finetuned LLM. While the finetuned LLM struggled especially with feasible demands, classifying them as infeasible, the RAG model showed a high accuracy and balance in both cases. These findings highlights how up-to-date and relevant contextual information may be more useful than inherent parametric knowledge for narrow and domain-specific RE tasks, such as feasibility analysis.

6.2 Future works

To further advance the finetuned pre-trained LLM, a dataset of higher quality and quantity is crucial. Data of higher quality would be well extracted requirements paired with an well formulated feasibility analysis by an expert. Additionally, as the finetuned LLM seemed to not learn from the synthetically created input-output pairs, more real-world data points would be needed. The real-world data would need to include examples of both infeasible and feasible demands for the LLM to learn how to differentiate between the two.

The pre-trained model selection was limited by the size of our GPU. With a larger

GPU, a model with more parameters could be chosen to finetune and possibly achieve higher performance. However, ensuring that the LLM would not draw from its inherent parametric knowledge would be harder and possibly requiring more extensive finetuning.

Exploring more advanced finetuning techniques could guide the model towards achieving higher performance. One such technique could be Reinforcement Learning from Human Feedback (RLHF), where the model learns directly from human input [64]. However, this would require extensive time from domain experts, which is not feasible for our study. A second technique could be to finetune the LLM with input-output pairs where the input contains text snippets instead of singled-out demands. This approach would provide the model with more natural context, helping it better interpret the intent and constraints behind the demand. This would allow the LLM make more informed judgments, which could be especially useful in cases where ambiguity arises.

One key area of improvement for the RAG model lies in the information retrieval quality of the embedding model. While embedding model achieved high accuracy, outperforming OpenAI's state-of-the-art model, future work could explore expanding the corpus with more data points, detailed technical specification, and historical project notes. This would give the semantic retriever a richer pool of context to draw from, further improving the RAG model.

As surfaced in the evaluation of the RAG, traceability and transparency is crucial for the experts being assisted by the RAG. To further improve these qualities, the RAG pipeline could incorporate direct access to the retrieved context used for information. As one expert highlighted in the questionnaire: *"You should try and program it so it point to the source of the information. This would assist further reading."* - Survey Participant. Such a mechanism would also allow the experts to easier validate the output.

6.3 Limitations and Delimitations

This section will discuss the limitations and delimitations which were deliberately scoped or imposed on the project.

6.3.1 Data

As high quality data was scarce and often could not be provided by Siemens Energy, we chose to generate synthetic data using GPT-4. This was done in cycle 1 to augment the existing dataset used for finetuning the pre-trained LLM with feasible demands. Additionally, in cycle 2, this was done to create customer demands for topics and their delivery option to craft a whole new dataset that could be used to finetune embedding models. While this method proved very useful in our case, it introduces potential errors since GPT-4 might not mimic the reasoning of domain experts perfectly.

6.3.2 Pre-trained models

In our study, we choose to use LLAMA-3.2-1B as the base for our finetuned pre-trained LLM solution, and the all-MiniLM-L6-v2 as our base for an embedding model to finetune in the RAG solution. To only include one base model for each solution, was a deliberate delimitation made in the project due to time and resource constraints. However, an extensive literature review was made to ensure that the chosen models were optimal for our solutions. Furthermore, our computational resources also set a limitation to our choice of pre-trained models. The computers that was made available to us by Siemens Energy had a GPU of 8GB, which made models with approximately more than 2 Billion parameters too large for our models. L

6.3.3 Language

One delimitation made in the project was choosing to only finetune on output-input pairs in English. Siemens Energy works with clients worldwide, and thus in many languages. In particular, Spanish, German, and French are the other languages which the corporation often has to analyze customer requirements in. Including these languages would include translating around 25% of the dataset to these languages. However, this would take time to do and significantly increase the training time of the models, and thus the delimitation to only finetune on English data was made.

6.3.4 Evaluation

The survey based evaluation focused on expert judgment from a limited number of a practitioners (20), which were a limitation. While insightful, the subjective nature of this type of evaluation introduces factors such as variability. The results may differ in broader organizational context.

7
Threats to validity

This chapter presents identified threats to validity of the study. The threats are divided into four categories: construct validity, conclusion validity, internal validity, and external validity.

7.0.1 Construct validity

Construct validity measures if the study has been able to correctly measure what the research questions intended to answer. The metrics used to evaluate the two models were relevance, correctness, and assistance. While these metrics capture important qualities of the models, there could still be other metrics that could be included as well. For example, coherence and context adherence are also relevant metrics to evaluate an LLMs performance. However, considering the purpose of the study, these metrics were deemed to fall under the metric 'assistance' because they are direct factors to this measurement. Further, no matter what metrics that were used, they would still be only serve as substitutes for more complex judgments. The use of expert opinion through the survey, mitigated this threat to some extent. Lastly, synthetic data was used for parts of the training. Even though the synthetic data closely resembled real-world data, it might not mimic the reasoning of domain experts perfectly.

7.0.2 Conclusion validity

Conclusion validity measures if the findings in the study are sufficiently supported by the data. The research made use of real-world data to develop the solutions, as well as the opinions of domain experts to evaluate the results, which strengthens the conclusions validity. Although, having more respondents of the survey would increase the statistical power of the results. While 20 respondents still gives a reasonable conclusion validity, it would always be beneficial with more. Further, the consulted domain experts provided some opinions that were conflicting, which suggests there are some subjectivity involved when judging the degree to which the models can assist in the process.

7.0.3 Internal validity

A possible threat to internal validity is the fact that all respondents and data are sourced from the same company, which could introduce selection bias. Furthermore, because the evaluation is a subjective measure made by domain experts, it could

introduce some confirmation bias. The experts might have prior experiences or preferences in the process, that could effect their responses in the survey. Additionally, the RAG showed a greater performance than the finetuned pre-trained LLM. This might not solely be due to the different architecture of the models, but could also be an effect of the data available at the company.

7.0.4 External validity

The study was conducted exclusively at Siemens Energy, which could limit the extent to which the findings are generalizable. For example, the exact nature of Siemens Energy's requirements engineering practices might differ from other companies. However, in the literature review it was found that this process is quite typical in the industry. If another company would like to replicate our solution in their process, the only limitation would be the amount and type of data they have available. The data needed for the solutions are a large corpus of product specifications and historical customer specifications with corresponding comments.

8

Conclusion

The current approach for the process of feasibility analysis is largely manual and relies heavily on the subjective opinions of domain experts. The process consists of iteratively comparing customer demands to standard product specifications, which has proved to be time-consuming, inconsistent and prone to human-errors. These limitations can lead to faulty requirements, which can be costly to fix during a later stage in the process.

In this study, the researchers aimed to explore how Large Language Models (LLMs) can assist in the process of feasibility analysis and help mitigate these limitations. The study investigated two different solutions and architectures for LLM; a finetuned pre-trained LLM and a Retrieval Augmented Generation (RAG) system. These models were evaluated in terms of their output's relevance, correctness and level of assistance to the employees. The study was done in collaboration with a big energy corporation, which made it possible to use real-world data and have domain experts evaluate the models.

The findings of the study revealed that the finetuned LLM was able to a high degree mimic the structure and language of the employees at the company. However, it typically lacked in the information which it provided, in terms of relevance and correctness. Specifically, it struggled with feasible requirements by frequently incorrectly identifying a gap to the product standards. In contrast, the RAG solution proved to be more reliable. It produced output that were traceable by providing technical references, and it was deemed to generate answers with better relevance, correctness and level of assistance to the employees. Furthermore, the RAG did not struggle with identifying whether a demand is feasible or not. An important finding of the study was that finetuning the embedding model, that is part of the RAG architecture, with external data from the company proved to be very effective. It was able to outperform state-of-the-art embedding models like OpenAI's text-embeddings-ada-002, in this specific context.

In conclusion, the study shows that LLMs, especially with a RAG architecture, could serve as an effective tool for assisting employees in the process of feasibility analysis. While there is lots of room for improvement, the models demonstrated a high potential in the process. Currently, the models are not able to are not able to completely take over the process but can instead stream-line and augment the work of the employees in this complex and technical domain.

Bibliography

[1] A. van Lamsweerde, "Requirements engineering in the year 00," in *Proceedings of the 22nd international conference on Software engineering - ICSE 00*, New York, NY, USA: ACM, 2000. DOI: 10.1145/337180.337184.

[2] N. G. Nahar, P. K. Wora, and S. Kumaresh, "Managing requirement elicitation issues using step-wise refinement model," *International Journal of Advanced Studies in Computers, Science and Engineering*, vol. 2, no. 5, p. 27, 2013.

[3] S. G. Gunda, *Requirements engineering: Elicitation techniques*, 2008.

[4] H. A. Bilal, M. Ilyas, Q. Tariq, and M. Hummayun, "Requirements validation techniques: An empirical study," *International Journal of Computer Applications*, vol. 148, no. 14, 2016.

[5] H. Ghanbari, J. Similä, and J. Markkula, "Utilizing online serious games to facilitate distributed requirements elicitation," *Journal of Systems and Software*, vol. 109, pp. 32–49, 2015, ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2015.07.017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121215001491.

[6] S.-C. Necula, F. Dumitriu, and V. Greavu-erban, "A systematic literature review on using natural language processing in software requirements engineering," *Electronics*, vol. 13, no. 11, p. 2055, Jan. 2024. DOI: 10.3390/electronics13112055.

[7] G. E. Mogyorodi, "What is requirements based testing," *Crosstalk: The Journal of Defense Software Engineering*, vol. 16, no. 3, p. 12, 2003.

[8] A. Tikayat Ray, B. F. Cole, O. J. Pinon Fischer, A. P. Bhat, R. T. White, and D. N. Mavris, "Agile methodology for the standardization of engineering requirements using large language models," *Systems*, vol. 11, no. 7, 2023, ISSN: 2079-8954. DOI: 10.3390/systems11070352. [Online]. Available: https://www.mdpi.com/2079-8954/11/7/352.

[9] A. Vogelsang and J. Fischbach, *Using large language models for natural language processing tasks in requirements engineering: A systematic guideline*, arXiv.org, May 2024. [Online]. Available: https://arxiv.org/abs/2402.13823.

[10] M. Krishna, B. Gaur, A. Verma, and P. Jalote, "Using llms in software requirements specifications: An empirical evaluation," *arXiv preprint arXiv:2404.17842*, 2024.

[11] K. Ronanki, "Enhancing requirements engineering practices using large language models," 2024.

[12] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in neural information processing systems*, vol. 36, pp. 10 088–10 115, 2023.

[13] GeeksforGeeks, *What is retrieval-augmented generation (rag)?* Accessed: 2023-10-10, 2023. [Online]. Available: https://www.geeksforgeeks.org/what-is-retrieval-augmented-generation-rag/.

[14] D. Zowghi and C. Coulin, "Requirements elicitation: A survey of techniques, approaches, and tools," in *Engineering and managing software requirements*, Springer, 2005, pp. 19–46.

[15] R. Pergl, "Feasibility study inputs based on requirements engineering," *Enterprise & Organizational Modeling and Simulation EOMAS 2010*, p. 121, 2010.

[16] A. Masoudifard, M. M. Sorond, M. Madadi, M. Sabokrou, and E. Habibi, *Leveraging graph-rag and prompt engineering to enhance llm-based automated requirement traceability and compliance checks*, 2024. arXiv: 2412.08593 [cs.SE]. [Online]. Available: https://arxiv.org/abs/2412.08593.

[17] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1706.03762.

[18] A. Q. Jiang, A. Sablayrolles, A. Mensch, *et al.*, *Mistral 7b*, 2023. arXiv: 2310.06825 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2310.06825.

[19] G. Team, T. Mesnard, C. Hardin, *et al.*, *Gemma: Open models based on gemini research and technology*, 2024. arXiv: 2403.08295 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2403.08295.

[20] M. AI, *Llama 3.2 1b - hugging face*, https://huggingface.co/meta-llama/Llama-3.2-1B, Accessed: 2025-03-27, 2024.

[21] H. Naveed, A. U. Khan, S. Qiu, *et al.*, *A comprehensive overview of large language models*, 2024. arXiv: 2307.06435 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2307.06435.

[22] Q. Dong, L. Li, D. Dai, *et al.*, *A survey on in-context learning*, 2024. arXiv: 2301.00234 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2301.00234.

[23] Y. Liu, H. He, T. Han, *et al.*, "Understanding llms: A comprehensive overview from training to inference," *Neurocomputing*, p. 129 190, 2024.

[24] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models.," *ICLR*, vol. 1, no. 2, p. 3, 2022.

[25] Y. Gao, Y. Xiong, X. Gao, *et al.*, *Retrieval-augmented generation for large language models: A survey*, 2024. arXiv: 2312.10997 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2312.10997.

[26] J. Barnard, *What are embeddings in nlp?* Accessed: 2023-10-10, 2023. [Online]. Available: https://www.ibm.com/think/topics/embedding.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, 2013. arXiv: 1301.3781 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1301.3781.

[28] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[29] GeeksforGeeks, *Ml | one hot encoding*, https://www.geeksforgeeks.org/ml-one-hot-encoding/, Accessed: 2023-10-15, n.d.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1810.04805.

[31] N. Reimers and I. Gurevych, *Sentence-bert: Sentence embeddings using siamese bert-networks*, 2019. arXiv: 1908.10084 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1908.10084.

[32] D. Cer, Y. Yang, S.-y. Kong, *et al.*, *Universal sentence encoder*, 2018. arXiv: 1803.11175 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1803.11175.

[33] M. Ormerod, J. Martínez del Rincón, and B. Devereux, "Predicting semantic similarity between clinical sentence pairs using transformer models: Evaluation and representational analysis," *JMIR Medical Informatics*, vol. 9, no. 5, e23099, 2021.

[34] X. Sun, Y. Meng, X. Ao, *et al.*, "Sentence similarity based on contexts," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 573–588, 2022.

[35] C. Mahachakir, *Understanding distance and similarity metrics in data analysis*, https://medium.com/@mahachakir/understanding-distance-and-similarity-metrics-in-data-analysis-aebdf85b920e, Accessed: 2023-10-15, 2019.

[36] S. Besrour, L. B. A. Rahim, and P. Dominic, "A quantitative study to identify critical requirement engineering challenges in the context of small and medium software enterprise," in *2016 3rd international Conference on Computer and Information Sciences (ICCOINS)*, IEEE, 2016, pp. 606–610.

[37] M. Kauppinen, M. Vartiainen, J. Kontio, S. Kujala, and R. Sulonen, "Implementing requirements engineering processes throughout organizations: Success factors and challenges," *Information and Software Technology*, vol. 46, no. 14, pp. 937–953, 2004, ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2004.04.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584904000692.

[38] M. Bano, "Addressing the challenges of requirements ambiguity: A review of empirical literature," in *2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*, 2015, pp. 21–24. DOI: 10.1109/EmpiRE.2015.7431303.

[39] O. J. Okesola, K. Okokpujie, R. Goddy-Worlu, A. Ogunbanwo, and O. Iheanetu, "Qualitative comparisons of elicitation techniques in requirement engineering," *Journal of Engineering and Applied Sciences*, vol. 14, no. 01, p. 2019, 2019.

[40] L. Liu, T. Li, and F. Peng, "Why requirements engineering fails: A survey report from china," in *2010 18th IEEE International Requirements Engineering Conference*, 2010, pp. 317–322. DOI: 10.1109/RE.2010.45.

[41] S. McLeod, "Feasibility studies for novel and complex projects: Principles synthesised through an integrative review," *Project Leadership and Society*, vol. 2, p. 100 022, 2021.

[42] M. Luisa, F. Mariangela, and N. I. Pierluigi, "Market research for requirements analysis using linguistic tools," *Requirements Engineering*, vol. 9, pp. 40–56, 2004.

[43] L. R. Wong, D. S. Mauricio, G. D. Rodriguez, *et al.*, "A systematic literature review about software requirements elicitation," *Journal of Engineering Science and Technology*, vol. 12, no. 2, pp. 296–317, 2017.

[44] R. Ankori, "Automatic requirements elicitation in agile processes," in *IEEE International Conference on Software-Science, Technology & Engineering (SwSTE'05)*, IEEE, 2005, pp. 101–109.

[45] K. Ronanki, B. Cabrero-Daniel, J. Horkoff, and C. Berger, "Requirements engineering using generative ai: Prompts and prompting patterns," in *Generative AI for Effective Software Development*, Springer, 2024, pp. 109–127.

[46] K. Ronanki, C. Berger, and J. Horkoff, "Investigating chatgpts potential to assist in requirements elicitation processes," in *2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, IEEE, 2023, pp. 354–361.

[47] K. VM, H. Warrier, Y. Gupta, *et al.*, "Fine tuning llm for enterprise: Practical guidelines and recommendations," *arXiv preprint arXiv:2404.10779*, 2024.

[48] A. Narimani and S. Klarmann, "Integration of large language models for real-time troubleshooting in industrial environments based on retrieval-augmented generation (rag),"

[49] M. Altendeitering, J. Pampus, F. Larrinaga, J. Legaristi, and F. Howar, "Data sovereignty for ai pipelines: Lessons learned from an industrial project at mondragon corporation," in *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, 2022, pp. 193–204.

[50] M. Staron, *Action research in software engineering*. Springer, 2020.

[51] T. Wolf, L. Debut, V. Sanh, *et al.*, *Transformers: State-of-the-art natural language processing*, `https://huggingface.co`, Accessed: 2025-03-27, 2020.

[52] R. Qasem, M. Hendi, and B. Tantour, "Alkafi-llama3: Fine-tuning llms for precise legal understanding in palestine," *arXiv preprint arXiv:2412.14771*, 2024.

[53] M. Qin, "The uniqueness of llama3-70b series with per-channel quantization," *arXiv preprint arXiv:2408.15301*, 2024.

[54] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, *Improving text embeddings with large language models*, 2024. arXiv: `2401.00368 [cs.CL]`. [Online]. Available: `https://arxiv.org/abs/2401.00368`.

[55] L. Xu, J. Lian, W. X. Zhao, *et al.*, *Negative sampling for contrastive representation learning: A review*, 2022. arXiv: `2206.00212 [cs.IR]`. [Online]. Available: `https://arxiv.org/abs/2206.00212`.

[56] Z. Yang, M. Ding, T. Huang, *et al.*, *Does negative sampling matter? a review with insights into its theory and applications*, 2024. arXiv: `2402.17238 [cs.LG]`. [Online]. Available: `https://arxiv.org/abs/2402.17238`.

[57] C. Yin and Z. Zhang, "A study of sentence similarity based on the all-minilm-l6-v2 model with same semantics, different structure after fine tuning," in *2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024)*, Atlantis Press, 2024, pp. 677–684.

[58] S. Sanjeev and A. Troynikov, "Embedding adapters," Chroma, Tech. Rep., May 2024, Accessed on July 8, 2024. [Online]. Available: `https://research.trychroma.com/embedding-adapters`.

[59] D. Wilianto and A. S. Girsang, "Automatic short answer grading on high school's e-learning using semantic similarity methods.," *TEM Journal*, vol. 12, no. 1, 2023.

[60] S. A. Salahudeen, F. I. Lawan, Y. Aliyu, *et al.*, "Hausanlp at semeval-2024 task 1: Textual relatedness analysis for semantic representation of sentences," in *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, 2024, pp. 188–192.

[61] J. Kurek, T. Latkowski, M. Bukowski, *et al.*, "Zero-shot recommendation ai models for efficient job–candidate matching in recruitment process," *Applied Sciences*, vol. 14, no. 6, p. 2601, 2024.

[62] OpenAI, "Introducing gpt-4.1 in the api," *OpenAI*, 2025, Accessed: 2025-05-08. [Online]. Available: `https://openai.com/index/gpt-4-1/`.

[63] R. Greene, T. Sanders, L. Weng, and A. Neelakantan, "New and improved embedding model," *OpenAI Blog. Available online: https://openai. com/blog/new-and-improved-embedding-model (accessed on 28 November 2023)*, 2022.

[64] Y. Bai, A. Jones, K. Ndousse, *et al.*, *Training a helpful and harmless assistant with reinforcement learning from human feedback*, 2022. arXiv: `2204.05862 [cs.CL]`. [Online]. Available: `https://arxiv.org/abs/2204.05862`.

A

Appendix 1

Table A.1: Survey Questions and Metadata

Survey Questions	**RQ(s)**	**Type**	**Num.**
Background of participants (demographic data)			
1. What is your role at Siemens Energy?	N/A	Text input	19
2. How much experience do you have in Requirements Engineering? (e.g., determining the feasibility of customer requirements and handling technical specifications)	N/A	Likert scale	20
3. How much experience do you have using LLMs (e.g., ChatGPT, Microsoft Copilot)?	N/A	Likert scale	20
Evaluation of responses			
4. Please evaluate the response generated by the RAG model based on the customer demand provided below. Demand: Commissioning, trial operation and start-up of the GT system are the primary responsibility of the Contractor in cooperation with the Purchaser/planner. The Contractor must ensure or check that all functions to be fulfilled with the participation of HLT are implemented and cooperate with their commissioning. **RAG response:** Commissioning, trial operation, and startup of the gas turbine system are included in Siemens Energy's scope, with dedicated tools and instruments provided by Siemens. Coordination with the Purchaser and involvement of HLT functions must be ensured during commissioning to verify all required functionalities are implemented and operational.			
4.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.2	Likert scale	20

4.2. **Correctness** (How correct and factual is the information in the response)	1.2	Likert scale	20
4.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.2	Likert scale	20
4.4. [OPTIONAL] Provide the reasoning for your answer.	1.2	Text Input	10
5. Please evaluate the response generated by the finetuned LLM based on the customer demand provided below. Demand: Commissioning, trial operation and start-up of the GT system are the primary responsibility of the Contractor in cooperation with the Purchaser/planner. The Contractor must ensure or check that all functions to be fulfilled with the participation of HLT are implemented and cooperate with their commissioning. **Finetuned LLM response:** Siemens Energy will provide technical advice and assistance during commissioning and start-up of the GT. During the commissioning phase Siemens Energy will lead the customer.			
5.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.1	Likert scale	20
5.2. **Correctness** (How correct and factual is the information in the response)	1.1	Likert scale	20
5.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.1	Likert scale	20
5.4 [OPTIONAL] Provide the reasoning for your answer.	1.1	Text Input	9

6. Please evaluate the response generated by the RAG model based on the customer demand provided below. Demand: The gas turbine heat and power plant is designed for fully automatic operation. Therefore, all components must be designed in such a way that they can be operated automatically by the main control system (I&C) in all load cases, during start-up and shut-down, in case of malfunctions, etc. **RAG response:** The main control system (typically Simatic PCS7) supports fully automatic operation of the gas turbine plant, including start-up, shut-down, load cases, and malfunction handling, ensuring all components are operable via automatic control. Redundancy options for CPUs and power supplies can be included to enhance reliability during faults. System capabilities like continuous operation at AC power failure (optionally included) and automatic start/stop switches also support fully automatic operation in all scenarios.			
6.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.2	Likert scale	20
6.2. **Correctness** (How correct and factual is the information in the response)	1.2	Likert scale	20
6.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.2	Likert scale	20
6.4 [OPTIONAL] Provide the reasoning for your answer.	1.2	Text Input	12
7. Please evaluate the response generated by the finetuned LLM based on the customer demand provided below. Demand: The gas turbine heat and power plant is designed for fully automatic operation. Therefore, all components must be designed in such a way that they can be operated automatically by the main control system (I&C) in all load cases, during start-up and shut-down, in case of malfunctions, etc. **Finetuned LLM response:** Some equipment in the plant can be controlled by the gas turbine control system but in some cases it makes financial and/or time sense to have additional HMIs for easier operator access.			
7.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.1	Likert scale	20
7.2. **Correctness** (How correct and factual is the information in the response)	1.1	Likert scale	20

5.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.1	Likert scale	20
7.4 [OPTIONAL] Provide the reasoning for your answer.	1.1	Text Input	10
8. Please evaluate the response generated by the RAG model based on the customer demand provided below. Demand: A 3-stage filter (M6, F9, E12) must be offered as an option. **RAG response:** A 3-stage filtration system with M6, F9, and E12 filter classes is not explicitly listed as a standard option; however, multiple 2-stage and 3-stage static and pulse-clean filter options are available for SGT600/700/750/800 models. Offering a 3-stage filter including E12 as an option would require verification and possible customization beyond the standard offerings described.			
8.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.2	Likert scale	20
8.2. **Correctness** (How correct and factual is the information in the response)	1.2	Likert scale	20
8.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.2	Likert scale	20
8.4 [OPTIONAL] Provide the reasoning for your answer.	1.2	Text Input	6
9. Please evaluate the response generated by the finetuned LLM based on the customer demand provided below. Demand: A 3-stage filter (M6, F9, E12) must be offered as an option. **Finetuned LLM response:** A 3 stage filter (F9 + E12) is offered as an option.			
9.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.1	Likert scale	20
9.2. **Correctness** (How correct and factual is the information in the response)	1.1	Likert scale	20

9.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.1	Likert scale	20
9.4. [OPTIONAL] Provide the reasoning for your answer.	1.1	Text Input	7
10. Please evaluate the response generated by the RAG model based on the customer demand provided below. Demand: *Redacted* **RAG response:** *Redacted*			
10.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.2	Likert scale	20
10.2. **Correctness** (How correct and factual is the information in the response)	1.2	Likert scale	20
10.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.2	Likert scale	20
10.4. [OPTIONAL] Provide the reasoning for your answer.	1.2	Text Input	8
11. Please evaluate the response generated by the finetuned LLM based on the customer demand provided below. Demand: *Redacted* **Finetuned LLM response:** *Redacted*			
11.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.1	Likert scale	20
11.2. **Correctness** (How correct and factual is the information in the response)	1.1	Likert scale	20
11.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.1	Likert scale	20
11.4. [OPTIONAL] Provide the reasoning for your answer.	1.1	Text Input	5

12. Please evaluate the response generated by the RAG model based on the customer demand provided below. Demand: *Redacted* **RAG response:** *Redacted*			
12.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.2	Likert scale	20
12.2. **Correctness** (How correct and factual is the information in the response)	1.2	Likert scale	20
12.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.2	Likert scale	20
12.4. [OPTIONAL] Provide the reasoning for your answer.	1.2	Text Input	8
13. Please evaluate the response generated by the finetuned LLM based on the customer demand provided below. Demand: *Redacted* **Finetuned LLM response:** *Redacted*			
13.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.1	Likert scale	20
13.2. **Correctness** (How correct and factual is the information in the response)	1.1	Likert scale	20
13.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.1	Likert scale	20
13.4. [OPTIONAL] Provide the reasoning for your answer.	1.1	Text Input	5
14. Please evaluate the response generated by the RAG model based on the customer demand provided below. Demand: *Redacted* **RAG response:** *Redacted*			
14.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.2	Likert scale	20

14.2. **Correctness** (How correct and factual is the information in the response)	1.2	Likert scale	20
14.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.2	Likert scale	20
14.4. [OPTIONAL] Provide the reasoning for your answer.	1.2	Text Input	8
15. Please evaluate the response generated by the finetuned LLM based on the customer demand provided below. Demand: *Redacted* **Finetuned LLM response:** *Redacted*			
15.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.1	Likert scale	20
15.2. **Correctness** (How correct and factual is the information in the response)	1.1	Likert scale	20
15.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.1	Likert scale	20
15.4. [OPTIONAL] Provide the reasoning for your answer.	1.1	Text Input	8
16. Please evaluate the response generated by the RAG model based on the customer demand provided below. Demand: *Redacted* **RAG response:** *Redacted*			
16.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.2	Likert scale	20
16.2. **Correctness** (How correct and factual is the information in the response)	1.2	Likert scale	20
16.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.2	Likert scale	20

16.4. [OPTIONAL] Provide the reasoning for your answer.	1.2	Text Input	6
17. Please evaluate the response generated by the finetuned LLM based on the customer demand provided below. Demand: *Redacted* **Finetuned LLM response:** *Redacted*			
17.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.1	Likert scale	20
17.2. **Correctness** (How correct and factual is the information in the response)	1.1	Likert scale	20
17.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.1	Likert scale	20
17.4. [OPTIONAL] Provide the reasoning for your answer.	1.1	Text Input	6
18. Please evaluate the response generated by the RAG model based on the customer demand provided below. Demand: *Redacted* **RAG response:** *Redacted*			
18.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.2	Likert scale	20
18.2. **Correctness** (How correct and factual is the information in the response)	1.2	Likert scale	20
18.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.2	Likert scale	20
18.4. [OPTIONAL] Provide the reasoning for your answer.	1.2	Text Input	7
19. Please evaluate the response generated by the finetuned LLM based on the customer demand provided below. Demand: *Redacted* **Finetuned LLM response:** *Redacted*			

19.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.1	Likert scale	20
19.2. **Correctness** (How correct and factual is the information in the response)	1.1	Likert scale	20
19.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.1	Likert scale	20
19.4. [OPTIONAL] Provide the reasoning for your answer.	1.1	Text Input	5
20. Please evaluate the response generated by the RAG model based on the customer demand provided below. Demand: *Redacted* **RAG response:** *Redacted*			
20.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.2	Likert scale	20
20.2. **Correctness** (How correct and factual is the information in the response)	1.2	Likert scale	20
20.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.2	Likert scale	20
20.4. [OPTIONAL] Provide the reasoning for your answer.	1.2	Text Input	6
21. Please evaluate the response generated by the finetuned LLM based on the customer demand provided below. Demand: *Redacted* **Finetuned LLM response:** *Redacted*			
21.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.1	Likert scale	20
21.2. **Correctness** (How correct and factual is the information in the response)	1.1	Likert scale	20

21.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.1	Likert scale	20
21.4. [OPTIONAL] Provide the reasoning for your answer.	1.1	Text Input	4
22. Please evaluate the response generated by the RAG model based on the customer demand provided below. Demand: *Redacted* **RAG response:** *Redacted*			
22.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.2	Likert scale	20
22.2. **Correctness** (How correct and factual is the information in the response)	1.2	Likert scale	20
22.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.2	Likert scale	20
22.4. [OPTIONAL] Provide the reasoning for your answer.	1.2	Text Input	4
23. Please evaluate the response generated by the finetuned LLM based on the customer demand provided below. Demand: *Redacted* **Finetuned LLM response:** *Redacted*			
23.1. **Relevance** (How relevant is the information provided in the response to the customer demand given)	1.1	Likert scale	20
23.2. **Correctness** (How correct and factual is the information in the response)	1.1	Likert scale	20
24.3. **Assistance** (How much would the response help in determining the feasibility of the demand)	1.1	Likert scale	20
24.4. [OPTIONAL] Provide the reasoning for your answer.	1.1	Text Input	5

25. [OPTIONAL] Do you have any other feedback?.	1.1 & 1.2	Text Input	8