Explainable AI for Decision Making

Applying Generative AI to Enhance Decision Making

Master's thesis in Data Science & AI

Fabian Kaneby, Johanna Norell

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Explainable AI for Decision Making

Applying Generative AI to Enhance Decision Making

FABIAN KANEBY, JOHANNA NORELL

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2025

Explainable AI for Decision Making
Applying Generative AI to Enhance Decision Making
FABIAN KANEBY, JOHANNA NORELL

Supervisor: Bettina Linder, Volvo Penta
Examiner: Mats Granath, Director - Complex Adaptive Systems M.Sc. Program

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

Cover: AI generated illustration of the human brain.

(Created with Microsoft Designer using the prompt "An abstract line drawing of AI and neural network with a human brain on white background (HEX: #ffffff)".

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Explainable AI for Decision Making
Applying Generative AI to Enhance Decision Making
FABIAN KANEBY, JOHANNA NORELL
Chalmers University of Technology

Abstract

This thesis examines the feasibility of using an AI system to support decision making processes in identifying potential root causes of quality issues in industrial and marine power systems. The AI system employs a Retrieval Augmented Generation (RAG) architecture, utilizing Large Language Models (LLMs).

The research investigates whether pre-trained LLMs, combined with a constructed database in the RAG framework, are sufficient to provide support in a highly specific domain context. It also explores the factors that influence user acceptance and trust in the AI system. The evaluation includes both quantitative metrics and qualitative user tests with domain experts.

The project was conducted in collaboration with Volvo Penta, a power solution provider, and all data collection and user testing were performed at the company. The findings suggest that the system can effectively retrieve and summarize historical data to aid in identifying the root causes of quality issues. Additionally, the study reveals that user satisfaction and trust of AI-driven insights are primarily influenced by the system's ability to explain its reasoning process for reaching conclusions.

Keywords: Generative AI, Large Language Models, Retrieval Augmented Generation, AI System, Explainable AI, Decision Support, Root Cause Analysis, Quality Issues

Acknowledgements

We would like to express our gratitude, to our examiner, Mats Granath, and our supervisor, Bettina Linder at Volvo Penta, for their invaluable support throughout this thesis project. It has been a great experience and an incredible learning process to combine academic research with real industrial impact.

Second, our appreciation goes to Volvo Penta for providing the opportunity to be a small part of the company's significant digitalization journey. The opportunities and challenges in this journey are tremendous, making this an extraordinarily interesting time to write this thesis. Additionally, we would like to acknowledge the support from Adam Wengrud and Himanshu Sahni for their technical assistance and knowledge sharing throughout our thesis project and to Jonas Trolle for his domain knowledge, visions and enthusiasm, guiding us throughout the thesis project.

Johanna Norell & Fabian Kaneby, Gothenburg, May 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AI	Artificial Intelligence
ANNS	Approximate Nearest Neighbor Search
CBOW	Continuous Bag of Words
FAISS	Facebook AI Similarity Search
LLM	Large Language Model
ML	Machine Learning
NLP	Natural Language Processing
PLM	Pre-trained Language Model
RAG	Retrieval-Augmented Generation
UX	User Experience

Contents

List of Acronyms **ix**

List of Figures **xiii**

List of Tables **xv**

1 Introduction **1**
1.1 Aim . . . 2
1.2 Research Questions . . . 2
1.3 Limitations . . . 3

2 Theory **5**
2.1 Large Language Models . . . 5
2.1.1 OpenAI GPT 4o-mini . . . 6
2.1.2 Llama 3 Instruct 70B . . . 6
2.1.3 Prompting techniques . . . 6
2.2 Retrieval-Augmented Generation . . . 7
2.2.1 Indexing . . . 7
2.2.1.1 Data Segmentation . . . 7
2.2.1.2 Embeddings . . . 8
2.2.1.3 Vector Storage . . . 8
2.2.2 Retrieval . . . 8
2.2.2.1 User Query . . . 9
2.2.2.2 Searching . . . 9
2.2.3 Generation . . . 9
2.3 Evaluation . . . 10
2.4 Human-Centered AI . . . 10
2.4.1 Explainable AI . . . 10
2.4.2 Useful & Usable AI . . . 11
2.4.3 AI-assisted decision making . . . 11

3 Methods **13**
3.1 Use case . . . 13
3.2 Data . . . 14
3.2.1 Data preprocessing . . . 15

3.2.2 Size of data sets . . . 15
3.3 AI System architecture . . . 15
3.3.1 Indexing . . . 16
3.3.1.1 Data Segmentation . . . 16
3.3.1.2 Embedding . . . 18
3.3.1.3 Vector Storing . . . 19
3.3.2 Retrieval . . . 19
3.3.3 Generation . . . 20
3.4 Explainability factors . . . 20
3.5 Evaluation . . . 21
3.5.1 Quantitative Evaluation . . . 21
3.5.1.1 Retrieval Component . . . 21
3.5.1.2 Generation Component . . . 22
3.5.2 Qualitative Evaluation . . . 23

4 Results **25**
4.1 Quantitative Evaluation . . . 25
4.1.1 Retrieval Component . . . 25
4.1.2 Generation Component . . . 27
4.2 Qualitative Results . . . 27
4.2.1 Searching for similar Quality Reports . . . 28
4.2.2 Analyzing the related Root Cause Analyses . . . 28
4.2.3 Importance of explainability factors . . . 29

5 Discussion **31**
5.1 Quantitative Evaluation . . . 31
5.1.1 Retrieval Component . . . 31
5.1.2 Generation Component . . . 32
5.2 Qualitative Evaluation . . . 33
5.2.1 Searching for similar Quality Reports . . . 33
5.2.2 Analyzing the related Root Cause Analyses . . . 34
5.2.3 Importance of explainability factors . . . 35

6 Conclusion **37**

Bibliography **39**

A Appendix 1 **I**
A.1 Prompt for feature engineering . . . I
A.2 Prompt for determining the most likely root cause . . . I

List of Figures

2.1 RAG Architecture 7

3.1 Illustration of the current process of how Quality Reports are handled within the company. 13
3.2 The desirable future state of how the AI System will enhance the process of handling Quality Reports within the company. 14
3.3 Description of the two distinct data sources and their relation. 15
3.4 Overview of the system architecture. Green components illustrate where AI have been used. 16
3.5 Overview of the feature engineering process. 17
3.6 Overview of embedding evaluation process. The same process was completed using *all-mpnet-base-v2*. 18
3.7 Illustration of the quantitative evaluation method. 22
3.8 Illustration of the generative evaluation method. 22

4.1 Accuracy of the quantitative evaluation on the retrieval component for all Quality Reports. 25
4.2 Accuracy of the quantitative evaluation on the retrieval component for Quality Reports from Volvo Penta only. 26
4.3 Accuracy of the retrieval component over different grouping sizes for all Quality Reports. 26

List of Tables

3.1 Overview of Data Sources and Features 14

4.1 Accuracy results across evaluation rounds 27
4.2 The users responses when evaluating the first case. 27
4.3 The users responses when evaluating the second case. 28
4.4 Responses from the participants when evaluating the components designed to enhance user satisfaction and trust. 29

1

Introduction

At the time of writing this report, we find ourselves in the midst of what is often referred to as the AI revolution. Although some may argue it is inaccurate to frame it this way, artificial intelligence (AI) and machine learning (ML) have existed since the early 1940s [1]. Nevertheless, several recent factors have significantly accelerated their adoption by a broader audience. Increasing availability and accessibility of data and computing power certainly play a pivotal role in the accelerating adoption of AI. Furthermore, it should be highlighted that the development and improvements in language models and generative AI have successfully showcased the power and elegance of AI to the general public.

As the AI revolution unfolds, several businesses have advanced their internal competencies and adoption in this domain. Data-driven decisions present a significant opportunity to make informed decisions, relying less on individual domain expertise. Ultimately, this opportunity includes scaling the number of decisions made without necessitating additional human resources.

Another key advantage of data-driven decision support systems is the potential of accelerating decision making. In large global companies, many long-standing decision processes remain highly complex, influenced by countless factors, therefore taking considerable time. For example, addressing faulty products in a large business context often involves prolonged lead times from issue detection to production changes. Potentially, AI systems can significantly reduce this delay by compiling and summarizing both historical and real-time data, enabling more efficient decision making.

However, there are obstacles and shortages in adopting AI techniques. Notably, the feasibility of the practical implementation of AI models and the and how to interact with their results are crucial considerations. Potential challenges in the process of developing these AI systems within industrial companies include data storage, data completion, access to data, technical adoption among other things. Lastly, human acceptance of AI systems is essential for the systems to be utilized once implemented. Thus, there is a strong need for case studies that address the feasibility of implementing AI systems in day-to-day business operations. This thesis aims to achieve precisely this, by implementing an AI model to support the decision making process at an industrial company. Data acquisition, model implementation and analysis have been conducted in coordination with Volvo Penta.

1.1 Aim

The aim of this thesis is to understand how emerging Large Language Model (LLM) techniques, specifically the Retrieval-Augmented Generation (RAG) architecture, can assist business users in decision making. Secondly, the aim is also to uncover and understand key factors for the user to accept and trust the RAG.

In order to achieve this, a proof of concept RAG is to be developed which guides users in deciding next course of action when handling reports of defect power systems. The RAG should be able to provide possible root causes based on historically similar cases. The goal of the RAG is to reduce time in the root cause investigation process related to a defect by enhancing this decision making process.

1.2 Research Questions

For manufacturing companies, analyzing claims and warranty cases is crucial for competitiveness, product quality perception and maintaining strong customer relationships. At Volvo Penta, a power solution provider, understanding quality issues is particularly complex due to several factors. First, their vast product range leads to a wide variety of potential issues. Secondly, the power solutions are often integrated into larger systems, where the system design is beyond the company's control. In these cases, installation and setup are critical factors affecting performance and related issues. Thirdly, operating in a global market, Volvo Penta encounters somewhat siloed claims processes, further complicating this process.

Although not all quality issues stem from production errors, investigating the root cause is sometimes essential to determine if a product change is necessary. Identifying the root cause is a time consuming process, often taking several months and requiring a thorough investigation. At Volvo Penta, this investigation is extensively documented, capturing extensive data and findings throughout the process. However, when faced with a new quality issue, navigating through this data has proven difficult. No systematic method for locating relevant data exists, often resulting in separate investigations for similar issues.

With the advancement of AI, particularly LLMs, Volvo Penta hypothesized that an LLM could enhance workflow efficiency. The ability of LLMs to manage large volumes of textual data and effectively summarize it has garnered attention as a potential tool for enhancing business processes.

However, traditional LLMs have shortcomings that must be addressed when used for critical business decisions. LLMs can be prone to hallucinations, generating fabricated information when lacking a direct answer instead of admitting uncertainty [2]. In highly specialized domains, they often lack the necessary domain-specific knowledge to provide accurate responses [2]. To combat these drawbacks, an evolution of the typical LLMS has been developed, referred to as Retrieval- Augmented Generation architecture [2].

In the RAG architecture, a retrieval component firstly retrieves information from a constructed database, providing relevant and accurate context to an LLM, minimizing the risk of hallucination in the output [2]. The database also acts as a knowledge base for the model, enhancing its utility in specific domains. Integrating AI architectures like RAG into daily operations presents challenges. Beyond design choices during development, user willingness to adopt and trust the system is crucial for its success [3].

As such, this thesis aims to answer the following questions.

- Can a RAG system effectively retrieve and summarize historical data to assist in identifying the root cause of engine quality issues?
- What factors influence user satisfaction and trust from AI-driven insights when identifying possible root causes?

1.3 Limitations

This thesis is written in collaboration with Volvo Penta, as such all data is limited to the Volvo Group and no data from different actors is used. Generalization from the model will therefore only be applicable on Volvo Group's domain and not outside. The data used for the thesis is limited to the time period 2018-01-01 to 2025-01-01.

2

Theory

This chapter presents the theoretical foundation of our AI system, beginning with its core technical components and continuing with a literature review on user satisfaction and trust in human-AI interaction. The first sections provide an overview of LLMs, including their underlying architecture, prompting techniques and limitations. We then introduce the Retrieval-Augmented Generation (RAG) architecture, which addresses the shortcomings of LLMs by integrating external knowledge retrieval. The chapter concludes with an exploration of human-centered AI, focusing on explainability, usability and decision support-factors critical to fostering user satisfaction and trust in AI-driven systems.

2.1 Large Language Models

Large Language Models are AI models designed to understand, process and generate text, making them a part of Natural Language Processing (NLP) [4]. Many recent breakthroughs in language models can be attributed to transformers, increased computational capabilities and large-scale training data being available [4].

In 2017, Google introduced the transformer architecture, which significantly advanced the way embeddings are generated and understood in NLP [5]. Unlike previous models that processed words sequentially, transformers uses a mechanism called self-attention, which allows the model to weigh the importance of each word in a sequence relative to others, regardless of their distance. This enables transformers to capture long-range dependencies in text more effectively [6].

Two prominent families of language models, Llama and Generative Pre-Training (GPT), are both built on transformer architectures [7, 8]. GPT serves as the foundation for OpenAI's models, while Llama is the foundation for Meta's models. LLMs like OpenAI's GPT models and Meta's Llama models consists of billions of parameters and are trained on vast datasets, enabling them to perform a variety of tasks, from drafting emails to serving as customer support agents [4]. However, like most AI, LLMs are dependent on the data they have been trained on. Its knowledge is limited to the information captured in their training data [4]. This limitation makes it difficult for LLMs to perform well in highly specialized domains where expert-level accuracy is required. LLMs are also prone to what is referred to as hallucinations, providing plausible sounding but factually incorrect information [9].

2.1.1 OpenAI GPT 4o-mini

As mentioned, OpenAI's LLMs are built upon their GPT architecture. GPT-4o is an autoregressive omni model. An autoregressive AI model predicts the next components from the previous sequence [6]. Following this, omni refers to a model that can take any combination of text, audio, image, and video as inputs and generate any combination of text, audio, and image as output [10]. It is trained on a combination of public available data and private data from partnerships [11]. GPT-4o-mini is a smaller version of GPT-4o, developed using model distillation. Model distillation is a technique where a smaller student model learns to mimic a larger teacher model by training on its outputs rather than the original dataset [12]. This process significantly reduces training costs across model families while maintaining comparable performance [13]. Neither GPT-4o-mini or GPT-4o are open source models.

2.1.2 Llama 3 Instruct 70B

Llama is an open source LLM developed by Meta, based on the transformer architecture and trained on data available to the public [8]. LLama differentiates from the original transformer architecture in various areas were it improves components including pre-normalization, the activation function and embeddings [8]. Llama 3 Instruct 70B is an instruction tuned model, intended for assistant-like chat, unlike pre-trained models which can be adapted for a variety of natural language tasks [14]. Instruction tuning refers to the fine-tuning of a language model on datasets consisting of input-output pairs framed as instructions [15]. The tuning has been done with supervised fine-tuning and reinforcement learning with human feedback [14].

2.1.3 Prompting techniques

Both OpenAI's GPT-4o-mini and Llama 3 Instruct are examples of pre-trained language models (PLMs). The use of PLMs has surged recently due to their exceptional performance, which stems from extensive training requiring substantial computational power and resources. Utilizing these models "out-of-the-box" is convenient and training models from scratch is unlikely to achieve comparable results. A method for enhancing task adaptation in pre-trained models is the application of *fine-tuning*. The *pre-train, fine-tune* approach leverages transfer learning, minimizing the need for labeled data, which is particularly beneficial in low-resource settings, such as domains or languages with limited annotated datasets [16]. However, the downsides of this approach, including the need for computational resources to fine-tune models, though less than if a model is trained from scratched and the necessity of understanding the architectures of the models being fine-tuned [16]. They further explore the field of *prompt-based learning*, which involves formatting prompts to guide models in producing desired outputs for NLP tasks. Using prompt engineering to enhance user input with predefined contextual prompts has demonstrated significant improvements of the generated outputs [17] [18].

This shift represents a transition from *pre-train, fine-tune* to a new paradigm *pre-*

train, prompt and predict [19]. This approach reformulates tasks to align with the model's existing knowledge rather than adapting the model itself, and thus requires a new skill set in *prompt engineering* [19]. Since the chosen prompt significantly influences the output of the LLM, identifying the most effective prompt to achieve desired results is of utmost importance.

2.2 Retrieval-Augmented Generation

To overcome the limited knowledgebase of LLMs and the risk of hallucinations appearing, the Retrieval Augmented Generation architecture has been developed. RAG systems combine pre-trained parametric memory, meaning that the knowledge is embedded in the parameters of the model itself, with a non-parametric memory, a database [20]. A typical RAG architecture, shown in Figure 2.1, consists of the mentioned parts, a vector database (non-parametric memory), an embedding model that encodes both the stored information and the user query and a pre-trained LLM (parametric memory) that generates responses based on the retrieved data.

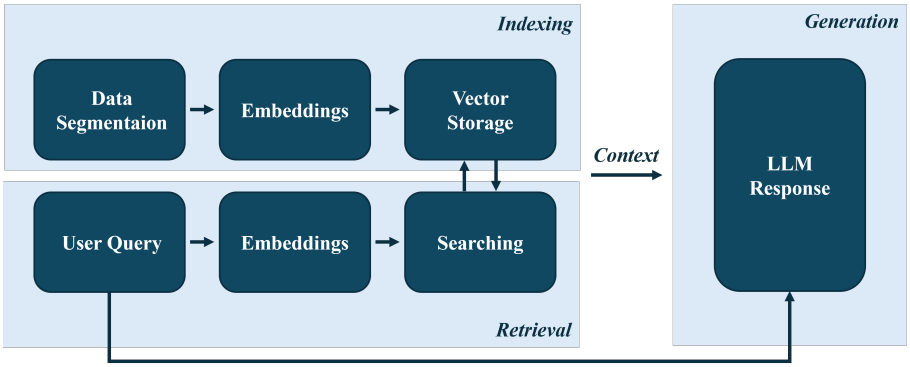

Figure 2.1: RAG Architecture

As illustrated in Figure 2.1, the RAG architecture consists of three main components: Indexing, Retrieval and Generation.

2.2.1 Indexing

Indexing involves creating a document database that serves as the information source for answering user queries. This process includes segmenting data into smaller parts to enhance search efficiency, embedding the information and storing it in vector indexes to facilitate database searches, as detailed by Kamath et al. in their work "Large Language Models: A Deep Dive" [16].

2.2.1.1 Data Segmentation

Segmenting data is crucial for optimizing the performance of the RAG system, as language models often experience significant degradation with long contexts [21]. Key steps include text pre-processing, chunking the data into smaller parts and

augmenting with metadata [16]. Further, Kamath et al. [16] highlight the importance of structuring information to leverage the RAG's capabilities and discuss several methods for incorporating information beyond simple document chunking. As part of this process, feature engineering can also be performed. This process aims to extract systematic features from raw data, transforming the data into a suitable format for a AI model [22]. It is further emphasized that a significant amount of time is often spent in this process, as it is essential to enable proper modeling of the features of the AI model.

2.2.1.2 Embeddings

Embeddings refer to a representation of data, for example text, images or sound, as numerical vectors in a continuous vector space. As the purpose of embeddings in the RAG architecture is to enable semantic similarity search, choosing a suitable embedding model is crucial to enable effective retrieval of information [16].

Representation of words as numerical vectors has for long been the foundation of NLP. One of the early embedding models, *Word2Vec*, was introduced by Google in 2013. *Word2Vec* uses two main training approaches, Continuous Bag of Words (CBOW) and Skip-gram [23]. CBOW predict the current word based on the context while Skip-gram predicts surrounding words given the current word. These techniques allow the model to encode meaningful word relationships, leading to the well-known algebraic operation on the word vectors below [23].

$$v_{\text{King}} - v_{\text{Man}} + v_{\text{Woman}} \approx v_{\text{Queen}}$$

Since then, word embeddings have evolved into sentence-level and contextual embeddings, enabling models to capture broader semantic meaning beyond individual words. Modern text embedding models like the ones available from Sentence Transformers and OpenAI create vector representation of sentences or paragraphs [24]. These models, built on transformer architectures, can encode entire paragraphs into dense, high-dimensional vector representations, preserving semantic meaning [24].

2.2.1.3 Vector Storage

Vector storage, handled by vector databases are used for storing and retrieving embeddings based on vector similarity. Similarity between vectors are defined by the distance, commonly calculated by Euclidian distance (L2 distance), cosine similarity or the inner product similarity [25]. Well known vector storing libraries includes Pinecone, Facebook AI Similarity Search (FAISS) and ChromaDB. Trade-offs between different choices often include search speed, scalability and the dynamic nature of the database [16].

2.2.2 Retrieval

The retrieval component of the RAG architecture aims to provide relevant knowledge to answer the user's query. The retrieval process involves a user querying the model, refining the query and searching the vector database for relevant documents.

2.2.2.1 User Query

The structure of a query, its clarity in expressing the user's intent and the semantic content it conveys all play a crucial role in retrieval performance [26]. Wang et al. [26] further outline methods for overcoming risks of poor querying, such as:

- **Query Rewriting:** Refines the query by rewriting it to better match documents.
- **Query Decomposition:** Retrieves documents based on derived sub-questions from the query.
- **Pseudo-documents Generation:** Generates a pseudo-document based on the query, embeds the answer retrieved from the pseudo-document and retrieves similar real documents based on the pseudo-answer.

These techniques aim to bridge the gap of semantic dissimilarity when the essence of the query aligns with the content in the stored vectors, as factors like grammatical dissimilarity can hinder searches in the RAG approach [16].

2.2.2.2 Searching

A fundamental assumption underlying the RAG architecture is that the information retrieved based on semantic similarity constitutes the relevant knowledge [16]. The RAG model retrieves the top ranked documents based on the similarity between the query and the documents, defined by the shortest distance between the vector representations.

Searching for the smallest distance can be done in many ways. Vector databases like FAISS and ChromaDB enables both Brute Force Search and Approximate Nearest Neighbor Search (ANNS) [25]. Brute force search computes the exact distances between all vectors and returns the top ranked results based on smallest distances between vectors, ensuring complete accuracy. For larger datasets, brute-force search becomes computationally inefficient, making ANNS a practical alternative. While ANNS accelerates the search process, it may sacrifice some precision in the results [25]. In most applications, the slight inaccuracy of ANNS is negligible compared to the significant speed advantage it offers [27].

2.2.3 Generation

After identifying relevant information from the RAG architecture's retrieval process, the final step is to generate output using the LLM. For successful generation, the retrieved information must be passed to the LLM in an appropriate format [16]. Research has examined how imperfect retrieval impacts the final generation. It can be noted that the retrieval precision is generally low, leading the LLM to often convey this imperfect information to the end user [28]. To address this issue, the internal knowledge of LLMs can be utilized to assess the reliability and coherence of the retrieved information, with the final answer based solely on consistent data [28].

2.3 Evaluation

RAG systems are typically evaluated based on the specific tasks they assist. The metrics used and the evaluation methods are determined by the task at hand and these metrics should effectively capture how well the system assists in achieving the task. While standard tasks such as question answering or summarization may use established metrics like Exact Match, F1 Score, or ROUGE, evaluation metrics can also be tailored to specific components of a system, such as retrieval or generation, by defining appropriate accuracy measures for each. In addition to task-specific evaluations, there have been efforts to assess RAG models objectively using standardized evaluation methods.

A possible way to evaluate the different functions of a RAG model is to focus on the performance of the two distinct parts of the model: *Retrieval Component* and *Generation Component* [2]. The retrieval performance can be evaluated with different metrics such as accuracy, precision, recall, mean reciprocal rank and mean average precision, with the goal of capturing relevancy of the information retrieved [29]. A key aspect of using any of these metrics is the requirement for an established ground truth. In some cases, the ground truth is subjective and depends on human evaluation. Evaluating the generation performance, the purpose is to consider factual correctness, readability and user satisfaction using metrics like BLEU, ROUGE and F1 Score [29].

2.4 Human-Centered AI

Beyond the technical aspects of AI systems, the human interaction with the technology can be evaluated. *Human-Centered AI* specifically examines the interaction and can be divided into two areas: *AI under human control* and *AI on the human condition* [30]. The first area focuses on the relationship between human and system control, whereas the second area emphasizes designing AI systems with a priority on explainability and interpretability to enhance human understanding [30].

Delving deeper into the aspect of human control within the learning process, we refer to *Human-in-the-loop machine learning*, which integrates human expertise into the technical learning framework. The degree of this integration enables us to evaluate whether the system or the humans retain sufficient control [3].

2.4.1 Explainable AI

Beyond the topic of control in the learning process, the human aspect of interaction and understanding of the AI model during deployment is referred to as *Explainable AI* [3]. Mosqueira-Rey et al. further emphasize that as the use of "black-box model" increases, where users have little to no insight into how decisions are made, the demand for greater transparency in the model's logic has become more pronounced [3]. They also highlight that explainable AI moves beyond simply explaining the model and how it functions, but also includes inherent intepretability from the model

design.

To understand the concept of explainable AI, five main factors should be considered: *understandability, comprehensibility, interpretability, explainability and transparency* [31]. Understandability, defined as the extent to which a human can grasp the model's decisions, is recognized as the most critical factor. Comprehensibility and interpretability relate to the model's ability to represent or explain knowledge and the learning process in a human-understandable manner. Transparency pertains to the model itself and its clarity; for instance, a linear regression model is generally more transparent than a neural network model. Transparency and explainability can be further analyzed through three components: simulability, decomposability and algorithmic transparency. Lastly, post-hoc explainability techniques can be employed to enhance interpretability. Examples of such techniques include text explanations, visual explanation and feature explanation, amongst other factors [3].

2.4.2 Useful & Usable AI

Exploring human interaction with AI reveals numerous factors that influence the adoption of AI models in business settings and their value in achieving specific goals. By considering both user needs and the ability to adopt solutions, we delve into the concepts of *Useful AI* and *Usable AI*. *Useful AI* emphasizes how effectively the AI model or system meets user needs and fulfills its intended purpose, while *Usable AI* focuses on the interface and the ease with which users can learn to interact with the tool [32]. Furthermore, advancements in understanding usage scenarios and user experience (UX) factors are discussed, emphasizing that these elements are key drivers of increased AI adoption [32]. Significant development is needed to understand how humans use these systems. Additionally, it is suggested to move beyond mere interaction to explore how AI systems and humans can transition from interaction to integration and collaboration, highlighting the need for further research in this area.

2.4.3 AI-assisted decision making

Deploying AI models can be categorized into two distinct use cases: *The AI model operates autonomously and makes final decisions* and *The AI model offers recommendations for action, leaving the final decision to a human* [33]. While the first option allows for complete automation of decision making, it may be infeasible for several reasons, and involving domain experts can help address these challenges by adding specialized knowledge [33]. For instance, determining medical diagnoses, evaluating creditworthiness, or making legal judgments are scenarios in which legal requirements and ethical considerations may prevent the fully autonomous deployment of AI models.

Elaborating on the second use case, where the AI model offer recommendations to humans, the area of *AI-assisted decision making* is defined. The final decision is then left to humans, which can choose to agree or disagree with the AI models suggestion [34]. To enhance human decision making with AI models, it is essential to optimize

joint decision outcomes. If users can form a mental model of their AI support, understanding when to trust or distrust predictions, they can effectively identify the model's error boundaries [33]. Calibrating user trust in the AI model's decisions can be achieved by displaying the confidence scores of the model's predictions [33]. However, this approach carries the risk of users becoming overly reliant on the model's predictions when the confidence score is very high.

The subjective nature of when to accept or reject assistance from AI in decision making has not yet been fully addressed in research [34]. Analyzing the relationship between algorithmic decision support and uncertainty, it has been found that, in the presence of irreducible uncertainty, uncertainty that cannot be resolved before an event occurs, human forecasters are preferred over algorithms [35]. Additionally, repeated observations of an algorithm making similar mistakes lead to increased aversion to using it as decision support [36].

3

Methods

Using the tools and techniques presented in the previous section, this section explores the practical use case and implementation at Volvo Penta. The subsections will follow the chronological order of events, from data exploration to final evaluation. We will refer to this practical case as designing an *AI System*, as we did not train a single specific model but aimed to design a comprehensive system for decision making using various AI techniques.

3.1 Use case

At the outset of this thesis, we transitioned from a vague idea of the AI system's use case to a clearer understanding of the decision making process we aimed to improve. This involved discussions with several domain experts, particularly the main stakeholder, a product quality manager. The process focused on understanding the current state, how decisions are currently made and the desired future state, the vision for how decisions could be assisted by AI. Through conversations with domain experts, we uncovered critical decision points with the highest potential for improvement and practical constraints for the systems deployment.

The use case with the largest potential was identified as aggregating and processing large amount of information from several sources. This would then help the user to find a possible root cause for a defect power system. At Volvo Penta, this is done by starting a Root Cause Analysis of the defect power system, where the defect is described in a Quality Report. An overview of the current state and vision of the future state can be found below in Figure 3.1 and 3.2.

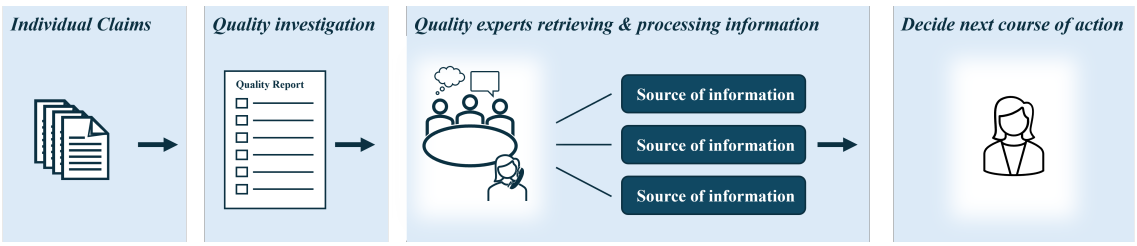

Figure 3.1: Illustration of the current process of how Quality Reports are handled within the company.

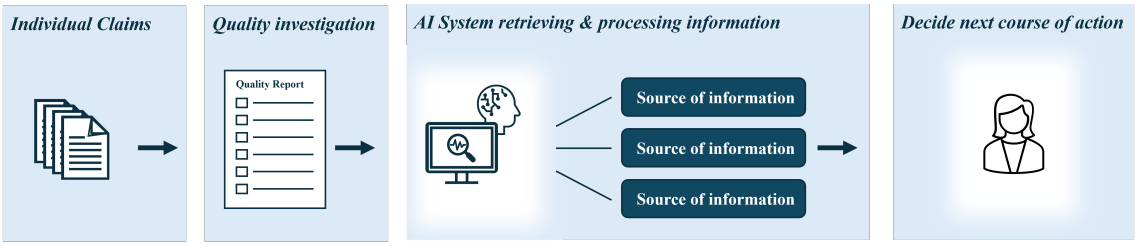

Figure 3.2: The desirable future state of how the AI System will enhance the process of handling Quality Reports within the company.

3.2 Data

Given the limited timeframe of this thesis, we determined that collecting or defining new data would be impractical. As a result, we relied on existing data for this thesis. The first stage involved assessing the available data to address the business needs. From all available data sources, we selected a subset that we deemed sufficient and valuable for enhancing the decision making process.

This resulted in two distinct datasets for the thesis: one related to the Quality Reports and one focused on investigating the Root Cause Analysis of those Quality Reports. From each dataset, we extracted various features, as presented in Table 3.1. In figure 3.3 the two data sets and their relation is illustrated.

Data source	Feature	Type
Quality Report	Feature 1	Textual
Quality Report	Feature 2	Textual
Quality Report	Feature 3	Textual
Quality Report	Feature 4	List<Categorical>
Root Cause Analysis	Feature 5	Textual
Root Cause Analysis	Feature 6	Textual
Root Cause Analysis	Feature 7	Textual
Root Cause Analysis	Feature 8	Textual
Root Cause Analysis	Feature 9	Categorical
Root Cause Analysis	Feature 10	Categorical
Root Cause Analysis	Target variable	Textual

Table 3.1: Overview of Data Sources and Features

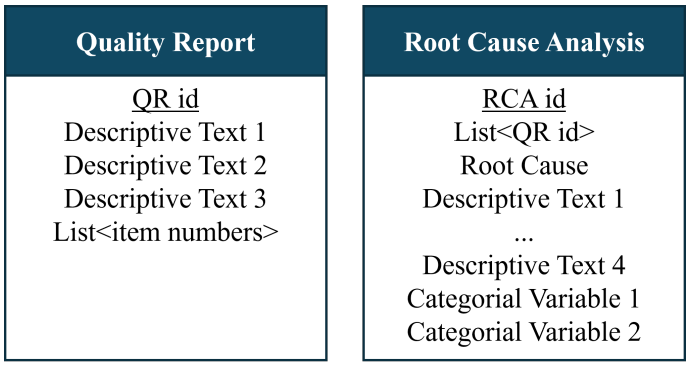

Figure 3.3: Description of the two distinct data sources and their relation.

3.2.1 Data preprocessing

Since the datasets used originate from distinct yet related sources, it was essential to ensure we had a sufficient and complete dataset to demonstrate the relationships of interest. As a result, we cleaned the data to remove any empty values in the target variables. Additionally, we refined the Root Cause Analysis dataset by eliminating any instances with empty references to Quality Reports. Lastly, we ensured that only instances of Quality Reports present in the Root Cause Analysis dataset were retained.

3.2.2 Size of data sets

After selecting the initial dataset, we needed to ensure it was large enough to identify relationships that could demonstrate how a data-driven approach could enhance the current process. We quickly realized that data from Volvo Penta alone was insufficient to build a robust model. Following consultations with domain experts, we decided to incorporate data from a sister company, Volvo Group Trucks Technology, to obtain the necessary dataset size. Although this decision might impact the relevance of some information in the database, the products from both companies are considered similar enough to provide adequate data for the proof of concept of this AI system, as this thesis aims to demonstrate.

The final data set used for building this system contained 6570 distinct instances of Quality Reports and 1699 distinct instances of Root Cause Analyses.

3.3 AI System architecture

After assessing the business needs and data availability, we drafted the overall system architecture to ensure that the data effectively supports the decision making process. The foundation of our solution is a RAG architecture, well-suited for handling advanced text similarity searches. After several iterations throughout the project, the final architecture was reached and is illustrated in Figure 3.4.

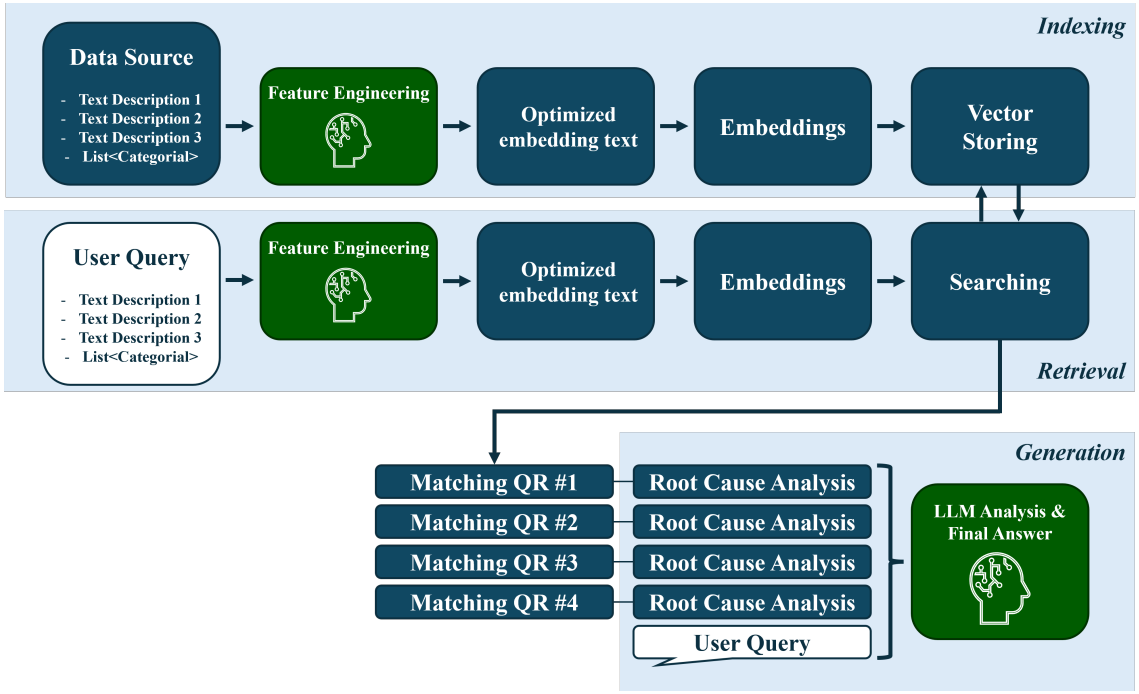

Figure 3.4: Overview of the system architecture. Green components illustrate where AI have been used.

As shown in Figure 3.4, the foundation of the system is a RAG architecture, although several modifications and adaptations have been made to transform the more traditional system architecture to better suit the specific needs of this project. The components of the three distinct parts, *indexing, retrieval*l and *generation*, will be described in the following sections.

3.3.1 Indexing

The following section will explain all the steps in the indexing process: *feature engineering, embedding* and *vector storing.*

3.3.1.1 Data Segmentation

The process of data segmentation was largely absent during the initial database design, as each Quality Report naturally served as a distinct data instance. However, we identified a strong need for extensive feature engineering to structure and standardize the historical Quality Report data.

We had a substantial amount of textual data to process and store for similarity searches. Since the majority of this data consisted of unstructured textual data, we recognized the need to extract more structured features. This was crucial for building our RAG architecture, where efficient search and retrieval are foundational components. As a result, we concluded that feature engineering was essential to extract the relevant information.

To enhance the feature engineering process, we decided to employ an LLM to process the input fields of the data. Collaborating with domain experts, we developed a strict prompt to extract relevant features from the text. These desired output features were defined to integrate both business and technical insights.

Firstly, we processed the data related to the Quality Reports. The data from the features, together with the prompt, was then sent to the LLM with a clear task to summarize the findings into a single text field. This process aimed to generate highly structured text with carefully selected information, which would later serve as the information database for our RAG system. The prompt template, slightly modified for privacy concerns from Volvo Penta, is found in appendix A.1.

In addition to optimizing the text field for embeddings, we prompted the LLM to assess the quality of the user input and the confidence level of its analysis. This aimed to be the foundation to evaluate how these features could be used to enhance the explainability and trustworthiness of the model. Figure 3.5 illustrates the full process of feature engineering.

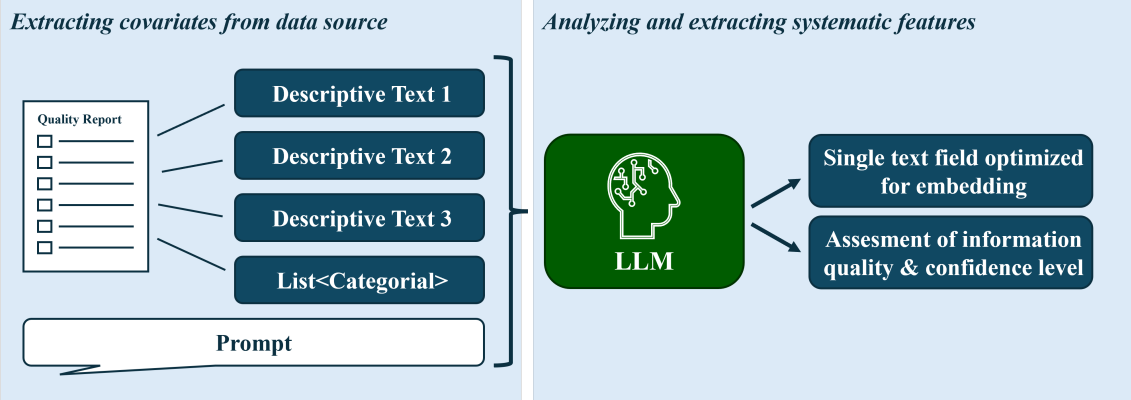

Figure 3.5: Overview of the feature engineering process.

Since these steps was critical for the model's performance in retrieving similar historical Quality Reports, we evaluated two different LLMs: GPT-4o-mini and Meta-Llama-3-7B-Instruct. These models were chosen due to their performance, ease of use and cost. Our goal was to compare the benefits of maintaining greater control over the model, as with the Llama model, versus using GPT-4o-mini.

We allowed the Llama model and GPT-4o-mini to perform feature engineering to compare their outputs across 40 Quality Reports. Additionally, we embedded the outputs generated by the two LLMs and analyzed their differences. A qualitative review of the outputs revealed minimal variation between the two models, as they produced highly similar texts.

To quantify this similarity, we calculated the Euclidean distance between the embedded texts from both LLMs, yielding an average similarity score of 0.948. The high score indicates a strong resemblance between the generated outputs. Given the

minimal performance differences observed, we decided to use GPT-4o-mini moving forward, as it offered faster computation time.

Further, we processed the Root Cause Analysis data set in the same manner by carefully crafting a prompt to extract relevant features from the six features available. In contrast to the processing of the Quality Report data, the goal of this process was rather to summarize and unify the data across different instances, to later enable the LLM to compare different possible Root Causes Analyses in a just way.

3.3.1.2 Embedding

To efficiently store and search within our database of historical Quality Reports, the next step after feature engineering was to embed the processed text. Since the quality of embeddings could significantly impacts the model's ability to find similar Quality Reports, we evaluated two different embedding models: *text-embedding-ada-002* from OpenAI and *all-mpnet-base-v2* from the Sentence Transformer Python framework. This allowed us to compare a high-performing API-based solution, though offering minimal insight or control over the embedding process, with an open-source alternative that provides greater flexibility and control.

To evaluate the embedding models, we sought a systematic approach to assess their ability to capture the essence of different textual inputs that convey the same semantics. We leveraged previous LLM processing of text fields describing Quality Reports using two different models (GPT-4o-mini and Llama) for 40 Quality Reports. These models produced slightly varied texts from the original descriptive inputs, yet aimed to represent the same semantics. We embedded these variations using both the *text-embedding-ada-002* and *all-mpnet-base-v2* as embedding model. Subsequently, we compared the top 5 matches each embedding model identified for a retrieval query.

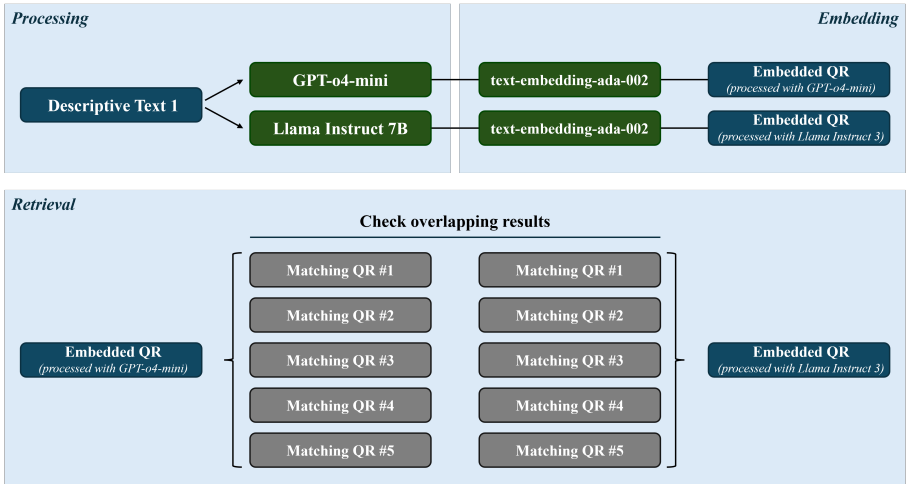

Figure 3.6: Overview of embedding evaluation process. The same process was completed using *all-mpnet-base-v2*.

The comparison of overlapping results measures how stable each embedding method is at handling minor variations in input. A high overlap in results indicates that the embedding model consistently captures key semantic features, whereas a lower overlap suggests greater sensitivity to small text differences. This helps determine which embedding method is more robust for retrieval tasks. *Text-embedding-ada-002* outperformed *all-mpnet-base-v2* with a mean score of 3.725/5 on the 40 Quality Reports and it was decided that embeddings using *text-embedding-ada-002* would be the choice for our system.

	GPT-4o-Mini & ada	GPT-4o-Mini & mpnet
Llama & ada	3.725/5	–
Llama & mpnet	–	2.850/5

Matrix 1: Average overlap of top 5 retrieval results when embedding similar texts

3.3.1.3 Vector Storing

To efficiently find similar embeddings produced in the previous step, proper storage was of essence. This may be critical to the system as the amount of data increases. As such, the FAISS vector storing library was used based on its performance and its ease of implementation. When creating the vector database, the *IndexFlatL2* index was used, which allows for brute-force euclidean distance search. As the size of our data set is relatively small, brute-force search could be used instead of the less compute intense ANNS.

3.3.2 Retrieval

The retrieval process closely mirrored the indexing process, albeit in a focused, single-instance manner based on the user query.

Rather than allowing the user to express themselves freely, we constrained their input to a format optimized for successful retrieval in the AI system. First, the user query was limited to filling out four fields, which corresponded to the four features used in the historical Quality Reports. Following, the same feature engineering steps was performed and the resulting text was embedded using the same embedding model used previously. This can be seen as a form of query rewriting where the aim is to refine the query to better match the stored documents.

We then employed a brute force euclidean distance search algorithm to retrieve the top ten matches. Each of the matches corresponded to a processed Quality Report. During the feature engineering process, each case was assigned one of four text quality labels, ranging from *Poor* to *Comprehensive*, by the LLM. The assigned text label was then used as a weight for the distance, refereed to as similarity scores, obtained from the brute force euclidean distance search, which resulted in a new weighted similarity score. This new score was subsequently used to narrow the results from 10 matches to four. The choice of retaining four potential matches was determined after iterative discussions with end users and domain experts. This number struck a balance between offering sufficient contextual matches, since some

cases that might not seem to match well could still be relevant and ensuring the information provided remained manageable for the user. Additionally, we retrieved the associated Root Cause Analysis data linked to the Quality Report data, which provided further contextual information for the final analysis and text generation.

3.3.3 Generation

In the generative part of the system, the relevant context was passed on from the retrieval section. This contextual information was provided to an LLM along with a specific prompt detailing how the LLM was to analyze the given context. Once the analysis was complete, the LLM returned its response to the user, consisting of recommendations for the next course of action. The prompt template, slightly modified for privacy concerns from Volvo Penta, used for this analysis and recommendation is found in appendix A.2.

3.4 Explainability factors

To explore how various factors influence user satisfaction and trust in the AI system, we maintained a continuous focus on incorporating explainability into the system. A key part of this approach, as previously mentioned, was having the LLM summarize the user input, express its confidence in the analysis during the feature engineering process and evaluating the quality of the textual input data. This strategy aimed to extract features from both the data and analysis process, which could later be presented to the user to improve the model's transparency and emphasize factors that could influence trust in the system's recommendations. A similar approach was applied during the final generative analysis phase, where the LLM was prompted to assess its confidence in both the analysis and the recommendation, along with providing two reasoning outputs explaining how it reached the final recommendation.

When retrieving the matching Quality Reports, the LLM was also asked to justify how and why these cases were considered similar. Lastly, the similarity score was transformed into one of three categorical variables with predefined threshold values, which were also presented to the end user.

In summary, the extracted features aimed at enhancing user satisfaction and trust were:

- Summary of user input:
  Output from the LLM during the feature engineering process.
- Text quality of user input:
  Output from the LLM during the feature engineering process.
- Confidence level of input data analysis:
  Output from the LLM during the feature engineering process.
- Similarity of matches:
  Combine score as describe in section 3.3.2

- Explaining why the matching Quality Reports are similar:
  Output from the LLM after finding matching Quality Reports.
- Considerations in how the recommended Root Cause Analysis was reached:
  Output from the LLM in the final analysis and output generation.
- Reasoning about alternative possible Root Cause Analyses:
  Output from the LLM in the final analysis and output generation.
- Confidence level of recommended Root Cause Analysis:
  Extracted by the LLM in the final analysis and output generation.

3.5 Evaluation

The evaluation consisted of a quantitative evaluation as well as a qualitative evaluation of the whole AI system. Each of the evaluation methods will be described in the following subsections.

3.5.1 Quantitative Evaluation

The quantitative evaluation, designed to provide an objective performance score for the system, was divided into two parts: one focused on the retrieval component and the other on the generation component.

3.5.1.1 Retrieval Component

For the quantitative retrieval evaluation, we aimed to assess how well the retrieval component of the system could find historical matching Quality Reports. To do this, we used historical Root Cause Analysis instances, each linked to several Quality Reports. All Root Cause Analyses which met the criteria of having between two and four related Quality Reports was selected. One of the linked Quality Reports was chosen as the input while the others acted as ground truth, as illustrated in Figure 3.7. If at least one of the ground truth Quality Reports was retrieved, it was counted as successful. If none of the retrieved Quality Reports were part of the ground truth, the retrieval was deemed unsuccessful. This was evaluated on the entire subset of data that met the specified criteria. The evaluation was conducted with retrieval of one historical Quality Report up to ten historical Quality Reports. By comparing successful and unsuccessful retrievals, an accuracy score could be calculated. As the retrieval component is dependent on the text in the Quality Reports, the text quality might change the results. Thus, a subset of the Quality Reports with text quality deemed as *Poor* was removed to see how it compared in accuracy. Further, we looked at Root Cause Analyses with groups of two, four and eight linked Quality Reports individually, to see how the accuracy changed depending the number of ground truths available.

Figure 3.7: Illustration of the quantitative evaluation method.

3.5.1.2 Generation Component

For the quantitative generation evaluation, we aimed to assess how well the generative component of the system could choose the correct Root Cause Analysis, focusing on instances linked to two Quality Reports, similar to the retrieval evaluation. Using cases with more than two linked reports would likely improve accuracy, as the LLM would have fewer Root Causes to consider. Therefore, focusing only on pairs represents a more restrictive scenario and the resulting accuracy can be viewed as a lower bound. One Quality Report was used as input, while the other was one of the four retrieved Quality Reports, ensuring that the Root Cause Analysis used as ground truth was among the considered Root Causes Analyses, as illustrated in Figure 3.8. This filtering reduced the data set to 88 pairs of Quality Reports. A generation was considered successful if the system selected the correct linked Root Cause Analysis, otherwise, it was considered unsuccessful. By comparing the number of successful and unsuccessful generations, we calculated an overall accuracy score.

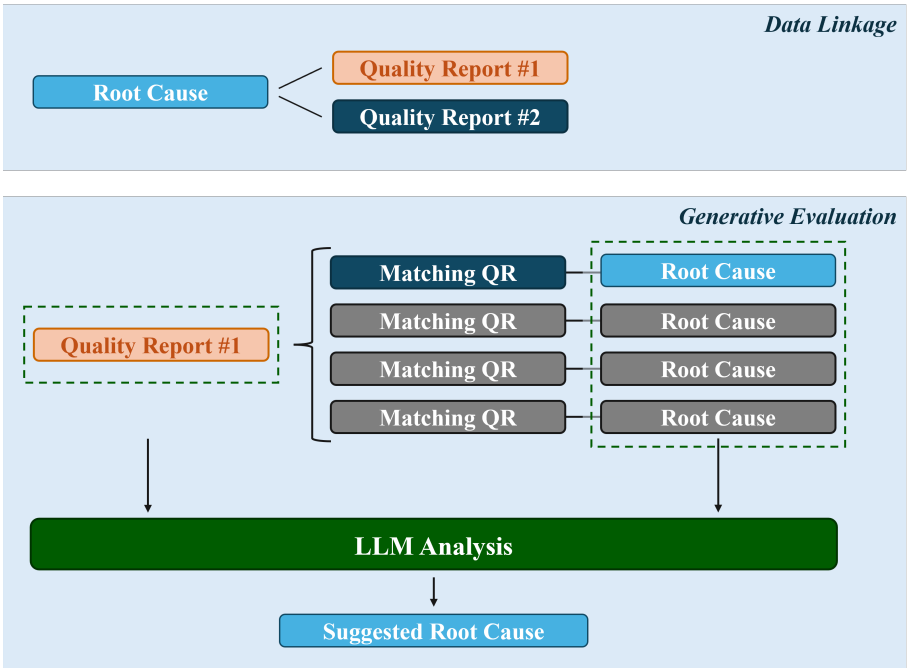

Figure 3.8: Illustration of the generative evaluation method.

3.5.2 Qualitative Evaluation

The qualitative approach consisted of user tests with domain experts. The evaluation aimed to evaluate both the system performance as well as its trustworthiness. Throughout the user tests, interview questions were asked about domain experts interaction and experience of the AI system.

Five domain experts, also potential future users of the system, were selected to evaluate it. Since the purpose of these user tests was to assess the system's future value for these potential users, no additional criteria were required for participant selection. All participants had over two years of experience in the field, consisting of two women and three men, aged between 29 and 61.

Each domain expert were given a brief explanation of the AI system and two Quality Reports used as input in the system. The Quality Reports were selected by our main stake holder who deemed them suitable for the test. During the user test, the following interviewing questions were asked to the user:

- Does the system accurately summarize the user input?
- How many of the matching Quality Reports are relevant to the user input?
- Is the recommended Root Cause Analysis plausible?
- Is the recommended Root Cause Analysis likely?
- Are any of the other Root Cause Analysis presented plausible?
- Would the recommended Root Cause Analysis be the first thing you investigate?

Once the two Quality Reports have been evaluated, the domain experts were asked questions related to the system as a whole and their experience of it.

- Do you trust the system?
- What specific part(s) of the system makes you trust it?
- Would a deeper knowledge in how the system works increase your trust in it?
- Would you use the system as it is?

The aim was to evaluate the overall performance and usefulness of the system, as well as investigate what aspects are important for user satisfaction and trust in AI systems.

4

Results

This chapter presents the results from each evaluation. First, the quantitative evaluation will be presented, covering both the retrieval and generative components. Second, the qualitative evaluation is outlined, including insights on system performance, explainability and trust factors.

4.1 Quantitative Evaluation

The following section presents the results from the quantitative evaluation on the retrieval and generation component.

4.1.1 Retrieval Component

When filtering the data for the quantitative evaluation criteria as described under 3.5.1.1, we were left with 1141 Quality Reports. Additionally, a filter on poor text quality was applied, leaving 1041 Quality Reports. The accuracy of both can be seen in Figure 4.1.

Figure 4.1: Accuracy of the quantitative evaluation on the retrieval component for all Quality Reports.

The accuracy of only Volvo Penta's Quality Reports were also calculated, in total 30 Quality Reports from Volvo Penta met the filtering criteria as described under 3.5.1.1. When applying the text quality filter 25 Quality Reports were left. The results are illustrated in Figure 4.2.

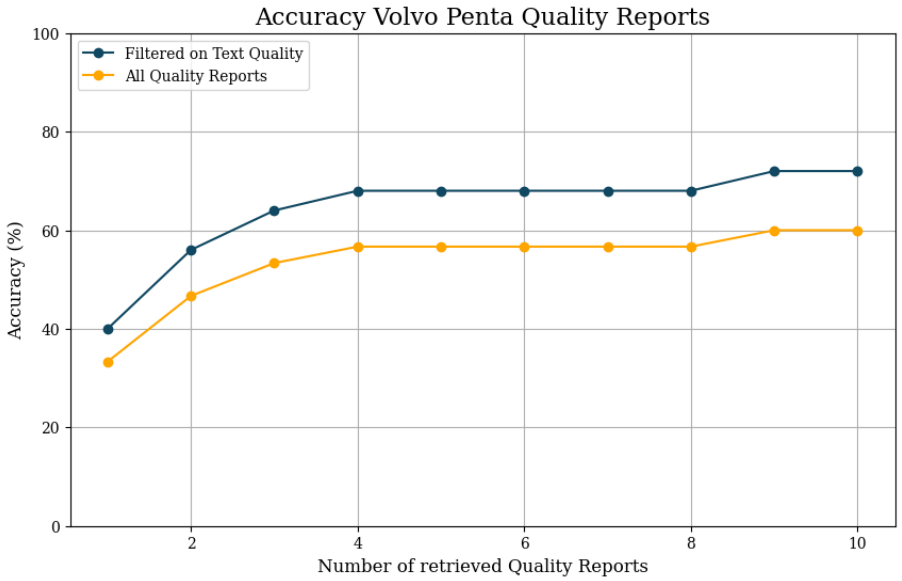

Figure 4.2: Accuracy of the quantitative evaluation on the retrieval component for Quality Reports from Volvo Penta only.

Additionally, different grouping sizes were evaluated to see how the accuracy changed depending on how many ground truth Quality Reports exists. The result where grouping size is 2, 4 and 8 is shown in Figure 4.3.

Figure 4.3: Accuracy of the retrieval component over different grouping sizes for all Quality Reports.

4.1.2 Generation Component

After filtering for the generative evaluation, described in 3.5.1.2 we were left with 88 Quality Report pairs. These where then evaluated and gave us an accuracy score for the generative component of the system and can be seen in Table 4.1. As the system is not completely deterministic, the evaluation was done three times on the same dataset. The system was consistent in 65 of the 88 cases, that is, it selected the same Root Cause Analysis for all iterations on 65 of the Quality Reports.

Evaluation	**Accuracy**
Iteration 1	76.14%
Iteration 2	72.73%
Iteration 3	73.86%
Mean	74.24%

Table 4.1: Accuracy results across evaluation rounds

By combining the results from the retrieval and generation components, we can calculate an overall accuracy for the system. The generative accuracy of 74.2% was measured under the assumption of 100% retrieval accuracy, meaning the relevant Quality Report was always retrieved. The retrieval accuracy with exactly two related Quality Reports was 49.2%, shown in Figure 4.3. Combining this retrieval accuracy with the generative accuracy, results in an overall system accuracy of 36.4%.

4.2 Qualitative Results

A summary of the participants responses to the two cases evaluated using the system are found in Tables 4.2 and 4.3. Participants are labeled P1, P2, P3, P4 and P5.

Evaluation Question	**P1**	**P2**	**P3**	**P4**	**P5**
Does the system accurately summarize the user input?	Yes	Yes	Yes	Yes	Yes
How many of the four matching Quality Reports are relevant to the user input?	3	4	3	4	0
Is the recommended Root Cause Analysis plausible?	Can't tell	Yes	Yes	Yes	No
Is the recommended root cause likely?	Can't tell	Yes	Yes	Yes	No
Are any of the other root causes presented plausible?	Yes	Yes	Yes	Yes	No
Would the recommended root cause be the first thing you investigate?	No	No	No	No	No

Table 4.2: The users responses when evaluating the first case.

Evaluation Question	P1	P2	P3	P4	P5
Does the system accurately summarize the user input?	Yes	Yes	Yes	Yes	Yes
How many of the four matching Quality Reports are relevant to the user input?	1	4	4	2	2
Is the recommended root causes plausible?	Yes	Yes	Yes	Yes	Yes
Is the recommended Root Cause Analysis likely?	Can't tell	Yes	Yes	Yes	Yes
Are any of the other root causes presented plausible?	N/A	Yes	Yes	Yes	No
Would the recommended root cause be the first you investigate?	N/A	Yes	No	Yes	No

Table 4.3: The users responses when evaluating the second case.

The following sections present the responses to all interview questions and highlights common themes that emerged in participants' reasoning divided into the distinct sections of the AI system: *searching for similar Quality Reports*, *analyzing the related Root Cause Analyses* and *importance of explainability factors*.

4.2.1 Searching for similar Quality Reports

All users agreed that the system accurately summarizes the user input. Four out of five participants specifically mentioned that this increased their confidence that the system shared their understanding of the problem description, while one participant did not comment on this aspect.

Continuing with the analysis of the matching Quality Reports, there was a general agreement on the relevance of most matches in the first case. However, in the second case, opinions were more divided. One observation was that participants less experience with the specific issue type tended to find more of the presented Quality Reports relevant, while those with deeper knowledge of the specific issue were more likely to consider some of the cases irrelevant.

4.2.2 Analyzing the related Root Cause Analyses

In the first case, three out of five participants found the recommended Root Cause Analysis plausible. Participant 1 expressed that they lacked sufficient information to assess its relevance to the problem description, while Participant 5 noted a mismatch between the contextual details in the problem description and those in the recommended Root Cause Analysis. In the second case, there was united agreement that the recommended Root Cause Analysis was plausible.

None of the participants expressed the view that the recommended Root Cause Analysis was plausible but unlikely. However, all participants emphasized that they would have needed additional information about the specific Root Cause Analysis before making a final judgment.

When asked whether the recommended Root Cause Analysis should be prioritized over the others, participants gave varied responses. Most emphasized that they preferred to review all available Root Cause Analyses to make their own informed decision. However, Participant 1 noted that if under time pressure, they would consider the recommended Root Cause Analysis first.

4.2.3 Importance of explainability factors

All participants agreed that they do not trust the system to make a final decision in any of the cases, but that they trust the information that the interface provides them with.

To evaluate more specifically what made the participants trust the system, we asked targeted questions about components designed to enhance user satisfaction and trust. Participants were asked whether each component increased their trust in the system. The result is presented below in Table 4.4.

Explainability component	**P1**	**P2**	**P3**	**P4**	**P5**
Summarizing the user input	Yes	Yes	Yes	Yes	Yes
Examining text quality of the user input	No	No	No	No	No
Examining confidence level in the input data analysis	No	No	No	No	No
Examining similarity score of matches	No	No	Yes	No	No
Explaining why the matching Quality Reports are similar	Yes	Yes	No	No	No
Explaining considerations in choosing the recommended Root Cause Analysis	No	No	No	No	No
Explaining alternatively considered Root Cause Analyses	Yes	Yes	No	No	No

Table 4.4: Responses from the participants when evaluating the components designed to enhance user satisfaction and trust.

All participants responded “yes” when asked whether the summary component contributed to their trust in the system. When elaborating, several participants valued

the component as a confirmation of alignment between the system and their understanding.

Regarding the components *"Explaining why the matching Quality Reports are similar"*, as well as *"Explaining alternatively considered Root Cause Analyses"*, participants who appreciated these features valued the ability to follow the model's reasoning. In contrast, those who responded negatively felt they could make these assessments themselves based on the information provided.

All participants agreed that they do not want a deeper knowledge about the logic behind the system. They did not believe it would enhance trust in any way.

All participants, except one, agreed they would use the system as it is if it became available to them. A noteworthy aspect is that the system's interface consolidates information from what is normally accessed through multiple systems when investigating these issues. This consolidation appears to enhance the system's usefulness, especially in comparison to the current workflow.

5

Discussion

In this chapter, we delve into a discussion of the quantitative and qualitative results obtained from our thesis. The aim is to interpret these findings within the context of the decision making processes they are designed to support. The discussion explores the performance of the system in the context of potential future users. We also explore the importance of explainability factors and their role in building trust in AI systems. This chapter serves as a critical reflection on the methodologies employed and offers insights into future directions for research and application.

5.1 Quantitative Evaluation

The following section discusses the results from the quantitative evaluation on the retrieval and generation components.

5.1.1 Retrieval Component

When examining the accuracy of all Quality Reports as well as the subset specific to Volvo Penta, a significant increase in accuracy up to the retrieval of four similar issues can be seen. Beyond this, the rate of improvement diminishes. This strengthens our choice of displaying the top four retrieved Quality Reports in the user interface, as displaying additional matches will not result in any significant improvement in the accuracy.

While the accuracy metric aims to give an objective measurement of the retrieval part of the system performance, it is important to note that this metric is influenced by certain characteristics of the data.

1. As the number of related Quality Reports ground truths increases, in theory, the accuracy should increase based on probability.

2. The related Quality Reports serving as the ground truth might not be the only relevant matching cases. In fact, more relevant Quality Reports linked to other Root Cause Analyses might exist. Retrieving them would in this evaluation count as an unsuccessful retrieval while the opposite might be true.

The first statement is partially confirmed in Figure 4.3 where it can be seen that the accuracy increases as related Quality Reports increase. However, it is challenging

to determine whether this improvement is proportional to the expected increase due to more ground truths or if other factors, such as data quality, influence the results.

The second statement is harder to interpret in our results. It can be argued that the achieved accuracy represents a lower bound of the true accuracy. As mentioned, retrieval is deemed unsuccessful if none of the related Quality Reports appear, even if the retrieved Quality Reports are directly relevant to the user input. Consequently, the true accuracy might be higher, but without domain experts manually assessing these cases, it remains unknown.

To conduct the quantitative evaluation, establishing a ground truth was essential, necessitating the previously outlined data selection process. While this approach was critical for ensuring the accuracy and reliability of the evaluation, it inadvertently led to the exclusion of a substantial amount of data. This loss primarily affected instances where only a single Quality Report was linked to a Root Cause Analysis, making it impossible to evaluate them using our methodology. Despite the necessity of this process, the reduction in available data highlights the challenges of balancing thoroughness with comprehensiveness in data-driven evaluations. The alternative of manually evaluating all instances with domain experts to obtain ground truths was not feasible due to time constraints, leaving the performance of these reports undetermined.

While the accuracy score provides a guidance in how well the system performs, a more suitable performance metric might be time saved for the user. However, no baseline for the current time spent in this process is available. From the qualitative evaluation with potential future users, it is indicated that the system would save them time and assist them in their work, but it is has not been possible to quantify the actual time save or performance increase. The accuracy metric might or might not mirror the true performance increase but the correlation between this accuracy score and actual time save is not necessarily high. A long-term study in which domain experts work in parallel, with one group using the system and another without it, would be necessary to establish a reliable baseline and accurately measure the system's performance gains.

5.1.2 Generation Component

As previously noted, the generation component is not entirely deterministic, meaning the same input may produce different outputs each time. To address this, we conducted multiple iterations of the evaluation and calculated the mean accuracy. Similar to the retrieval accuracy, the generation accuracy of 74.24% can be considered a lower bound. This is due to the fact that a more suitable Root Cause Analysis for the user input might exist beyond the perceived ground truth. Without comprehensive manual evaluation by domain experts, this remains uncertain.

Similar to retrieval evaluation, the number of related Quality Reports to a Root Cause Analysis affects generation accuracy, albeit differently. With more related Quality Reports, the likelihood of considering fewer Root Cause Analyses increases. For instance, if three out of four retrieved Quality Reports share the same Root

Cause Analysis, the generation component examines only two distinct Root Cause Analyses, potentially enhancing the chance of selecting the correct one. As we only tested Root Cause Analyses with exactly two related Quality Reports, this was not confirmed. The logic does however further the idea that the achieved accuracy serves as a lower bound for the true accuracy.

The same discussion regarding the accuracy metrics correlation to general performance increase written under 5.1.1 applies here too. Correlation is not necessarily high.

5.2 Qualitative Evaluation

As is often the case in many RAG applications, this evaluation has yielded noteworthy results. While there are numerous aspects that fall outside the scope and structure of our evaluation in this thesis, certain areas and patterns have emerged that deserves attention. Although the limited number of participants prevents us from drawing definitive general conclusions, the results can be regarded as indicative. This part of the discussion will focus on the following distinct parts of the AI system: *searching for similar Quality Reports*, *analyzing the related Root Cause Analyses*, *importance of explainability factors* as well as a *final reflection about the system as a whole and its drawbacks*.

5.2.1 Searching for similar Quality Reports

The first component of this part of the system involves the AI summarizing the user's input. Results indicate that providing a summary, rather than omitting it, helps build trust in the system. This is achieved by confirming the user's problem description, allowing them to verify that the AI has accurately understood the issue without introducing hallucinations or irrelevant content. However, the perceived importance of this component appears to vary between individuals. It should also be noted that the problem descriptions used in both test cases were relatively brief, which means the full potential of this component may not have been fully explored.

Based on user input, the system searches for similar historical Quality Reports and identifies relevant matches with reasonable accuracy. Although we did not establish a clear benchmark for evaluating the relevance of these matches, we relied on subjective assessment guided by our understanding of the data and context. Given the unstructured nature of the textual data in this domain, where exact matches are rarely expected, it is encouraging that users perceived the suggested cases as relevant. However, we cannot determine with certainty whether these were the most relevant matches, as it would be impractical to manually evaluate all 6,500+ historical Quality Reports for comparison.

Two participants mentioned the idea of "garbage in, garbage out", pointing out that the system's effectiveness largely depends on the quality of the problem description provided by the user. In addition, throughout the development process,

we considered the quality of the data within the underlying database. The system's performance is inherently limited by the quality of the information available for analysis. Another challenging scenario arises when the system encounters cases with no suitable historical Quality Reports. Currently, it suggests four suboptimal matching Quality Reports and often recommends a Root Cause Analysis from an unrelated field. Although establishing a minimum similarity threshold for displaying matching Quality Reports could mitigate this issue, a more effective strategy would involve deriving new conclusions and proposing innovative solutions. By leveraging historical problem-solving patterns alongside technical specifications of components and power solutions, it would be intriguing to assess the system's ability to generate innovative solutions.

Our reflections on data quality centered on two key aspects. The first pertains to the volume and comprehensiveness of historical data, addressing the issue of occasionally presenting users with suboptimal matching cases, as discussed above. This consideration influenced an early decision to expand the data source from only Volvo Penta data to include data from Volvo Group Trucks Technology data as well. However, several evaluations later indicated that the data from Volvo Group Trucks Technology was not always perceived as relevant. The second aspect relates to the quality of individual data entries. Many of the historical Quality Reports in the database contained incomplete or vague information. Regardless of how well the user's input is formulated, low-quality data provides a weak foundation for identifying meaningful matches.

Finally, it is important to acknowledge that, at the outset of this project, we lacked the experience and perspective to critically assess the quality and relevance of the data. Due to time constraints and unfamiliarity with the dataset and its context, we initially underestimated the significance of this aspect. Only toward the end of the thesis work did we begin to fully grasp how crucial data quality is to the performance and reliability of the system.

5.2.2 Analyzing the related Root Cause Analyses

In general, participants' reasoning regarding the relevance of the presented Root Cause Analyses was closely linked to the relevance of the matching historical Quality Reports. This supports the underlying logic of using the similarity of Quality Reports as a basis for identifying possible Root Causes Analyses.

The evaluation of both the relevance of all suggested Root Cause Analyses and the system's final recommendation suggests that multiple valid solutions to a problem often exist and that the system is capable of identifying several relevant possible solutions. However, when it comes to whether the recommended Root Cause Analysis should be prioritized over the others, responses indicate some level of skepticism. Participants appeared less inclined to fully trust the system's recommendation, particularly when they had the expertise to critically assess the alternatives themselves.

This suggests that the recommendation of a single Root Cause Analysis is considered less valuable than the broader ability to explore several plausible options. Trust in

the recommendation seems highly dependent on the specific problem description and how well both the user and the system have managed to capture the nature and context of the issue. If the user is better at capturing the nature and context, they tend to rely less on the system recommendation.

5.2.3 Importance of explainability factors

We have explored *AI on the human condition* to assess the interpretability of the model. Although LLMs differ significantly from other "black-box models", understanding their specific decision making processes remains challenging. While we did not attempt to examine all components of *Explainable AI*, we aimed to incorporate *understandability*, *explainability* and *transparency* into certain aspects of the model. This focus is particularly relevant given that we did not train a model ourselves but utilized an existing one within our system.

As previously discussed, the provided summary of the user input serves its purpose as a form of verification. Based on our results, it is indicated that participants view this as a necessity rather than a trust-building component. The expected behavior is for it to summarize accurately, thus not increasing trust but meeting expectations. However, it is important to note that this component may not have been fully evaluated in the pre-designed test cases used during the qualitative assessment. Participants were limited to the provided input, which was neither particularly long nor complex, making it relatively easy to summarize. A more targeted evaluation of this component, using more complex or user-generated input, could potentially reveal its full value and explore its purpose in building trust.

When examining the factors that helped build trust in the AI system, the results diverged to some degree. However, a common theme among several participants was that trust was more strongly influenced by the reasoning components we presented, rather than by decisions made by the LLM, such as determining a specific confidence level. It appears that participants were hesitant to trust these specific decisions, instead placing more value on the reasoning behind the systems' decision making process than on the decisions themselves.

It was also evident that some users valued these factors less than others. This seems to be closely related to the users' specific needs and expectations regarding what the system can and should provide. The results from evaluating the explainability components revealed patterns similar to those observed when evaluating the search for matching cases. Users who sought broader matches tended to value these components more than those looking for very specific matches and relationships. The findings also suggest that users with a narrower search strategy may have a clearer approach to analyzing and identifying historical Quality Reports. It is possible that they pay less attention to the reasoning components because they already have a structured method for identifying these aspects themselves. Additionally, they may be more accustomed to performing this process without the aid of an AI system, making them less likely to engage with the AI components and assistance. Another factor to consider is the system's performance and how it may influence user behav-

ior. If the system does not provide sufficiently relevant information, the perceived usefulness will naturally be lower.

An intriguing observation from the evaluation of trust-building factors is that none of the participants expressed a desire for deeper knowledge about the system's back-end logic. The common justification for this was the preference to trust the system as it is, without needing further understanding of its inner workings. This raises important considerations about the choices made regarding the logic that significantly influences the model's outcomes and performance, especially given that users are indifferent to these mechanisms. In our system, a crucial decision was how to prioritize which Quality Reports were presented to end users. This was determined by a relevance score, which combined the FAISS similarity score with a weight from the text quality. Since this score fully dictated which Quality Reports were ultimately shown to users, the choice of logic is extremely important. Our reflection, underscored by the users' lack of interest in this logic, is that involving the right people in making these decisions is vital. It requires substantial domain knowledge to ensure the decisions are made correctly.

While the usefulness of the AI system has been extensively discussed in previous sections, the concept of *Usable AI* can be analyzed from the questions addressing the overall perception of the system. Although this thesis did not focus on the front-end application and interface, it is evident that a significant value of this solution lies in integrating the gathered information into a single interface. This integration alone can be seen as a time saver and potential performance improvement of the users' analysis.

Finally, it is important to emphasize that all participants in the evaluation agreed they would not trust the system to make decisions without human involvement. This underscores that the system is designed to assist in decision making rather than fully automate the process.

6

Conclusion

It is indicated that a RAG system effectively can retrieve and summarize historical reports to assist in identifying the root cause of engine quality issues. With a retrieval accuracy of 49.2% and a generation accuracy of 74.2%, combined with insights from domain expert interviews, the system demonstrates its usefulness. However, the relationship between these accuracy metrics and the time saved for users remains unknown. A meaningful baseline for the current time expenditure has not been established and an evaluation of the time spent using the system has yet to be conducted. The key component of the RAG architecture used in this system is its database; however, this database also presents a significant drawback. The system is heavily reliant on it and when relevant information is absent, the resulting output often becomes irrelevant.

Components that appear to influence user satisfaction and trust in AI-driven insights are primarily the reasoning components that explain the system's process for reaching its conclusions. Providing an initial summary of the AI system's understanding of the input allows users to verify that it aligns with their understanding of the problem. This seems to be a hygiene factor in AI systems rather than a means of building trust. Confidence statements from the AI system do not seem to affect user trust; instead, users tend to make that assessment themselves. User satisfaction with the AI system appears to depend on its performance compared to current methods of working.

Bibliography

[1] Peter, N., 2021. Artificial Intelligence: A Modern Approach, Global Edition. Pearson Education Limited.

[2] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H. and Wang, H., 2023. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2.

[3] Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J. and Fernández-Leal, Á., 2023. Human-in-the-loop machine learning: a state of the art. Artificial Intelligence Review, 56(4), pp.3005-3054.

[4] Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N. and Mian, A., 2023. A comprehensive overview of large language models. arXiv preprint arXiv:2307.06435.

[5] Worth, P.J., 2023. Word embeddings and semantic spaces in natural language processing. International journal of intelligence science, 13(1), pp.1-21.

[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

[7] Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I., 2018. Improving language understanding by generative pre-training.

[8] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F. and Rodriguez, A., 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

[9] Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W. and Do, Q.V., 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. arXiv preprint arXiv:2302.04023.

[10] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. and Krueger, G., 2021, July. Learning

transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PmLR.

[11] Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A.J., Welihinda, A., Hayes, A., Radford, A. and Mądry, A., 2024. Gpt-4o system card. arXiv preprint arXiv:2410.21276.

[12] Liu, C., Tao, C., Liang, J., Feng, J., Shen, T., Huang, Q., & Zhao, D. (2023, December). Length-Adaptive Distillation: Customizing Small Language Model for Dynamic Token Pruning. In Findings of the Association for Computational Linguistics: EMNLP 2023 (pp. 4452-4463).

[13] Muralidharan, S., Turuvekere Sreenivas, S., Joshi, R., Chochowski, M., Patwary, M., Shoeybi, M., Catanzaro, B., Kautz, J. and Molchanov, P., 2024. Compact language models via pruning and knowledge distillation. Advances in Neural Information Processing Systems, 37, pp.41076-41102.

[14] AI@Meta. (2024). Llama 3 Model Card. Available at: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md (Accessed: 9 April 2025).

[15] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A. and Schulman, J., 2022. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35, pp.27730-27744.

[16] Kamath, U., Keenan, K., Somers, G. and Sorenson, S., 2024. Large Language Models: A Deep Dive. Springer.

[17] Zhang, X., Talukdar, N., Vemulapalli, S., Ahn, S., Wang, J., Meng, H., Murtaza, S.M.B., Leshchiner, D., Dave, A.A., Joseph, D.F. and Witteveen-Lane, M., 2024. Comparison of prompt engineering and fine-tuning strategies in large language models in the classification of clinical notes. AMIA Summits on Translational Science Proceedings, 2024, p.478.

[18] Zhou, H., Li, M., Xiao, Y., Yang, H. and Zhang, R., 2024. LEAP: LLM instruction-example adaptive prompting framework for biomedical relation extraction. Journal of the American Medical Informatics Association, 31(9), pp.2010-2018.

[19] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H. and Neubig, G., 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM computing surveys, 55(9), pp.1-35.

[20] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.T., Rocktäschel, T. and Riedel, S., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems, 33, pp.9459-9474.

[21] Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F. and Liang, P., 2024. Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics, 12, pp.157-173.

[22] Zheng, A. and Casari, A., 2018. Feature engineering for machine learning: principles and techniques for data scientists. " O'Reilly Media, Inc.".

[23] Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[24] Günther, M., Ong, J., Mohr, I., Abdessalem, A., Abel, T., Akram, M. K., ... Xiao, H. (2023). Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. arXiv preprint arXiv:2310.19923.

[25] Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.E., Lomeli, M., Hosseini, L. and Jégou, H., 2024. The faiss library. arXiv preprint arXiv:2401.08281.

[26] Wang, X., Wang, Z., Gao, X., Zhang, F., Wu, Y., Xu, Z., Shi, T., Wang, Z., Li, S., Qian, Q. and Yin, R., 2024. Searching for best practices in retrieval-augmented generation, pp. 17716-17736, arXiv preprint arXiv:2407.01219.

[27] Szilvasy, G., Mazaré, P.E. and Douze, M., 2024. Vector search with small radiuses. arXiv preprint arXiv:2403.10746.

[28] RAG, A. Knowledge Conflicts for Large Language Models. Submitted to ACL Rolling Review, December 2024. https://openreview.net/forum?id=WVDzLJMd7H

[29] Yu, H., Gan, A., Zhang, K., Tong, S., Liu, Q. and Liu, Z., 2024, August. Evaluation of retrieval-augmented generation: A survey. In CCF Conference on Big Data (pp. 102-120). Singapore: Springer Nature Singapore.

[30] Yang, S.J., Ogata, H., Matsui, T. and Chen, N.S., 2021. Human-centered artificial intelligence in education: Seeing the invisible through the visible. Computers and Education: Artificial Intelligence, 2, p.100008.

[31] Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R. and Chatila, R., 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information fusion, 58, pp.82-115.

[32] Xu, W., 2019. Toward human-centered AI: a perspective from human-computer interaction. interactions, 26(4), pp.42-46.

[33] Zhang, Y., Liao, Q.V. and Bellamy, R.K., 2020, January. Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In Proceedings of the 2020 conference on fairness, accountability, and transparency (pp. 295-305).

[34] Taudien, A., Fügener, A., Gupta, A. and Ketter, W., 2022. The effect of AI advice on human confidence in decision-making.

[35] Dietvorst, B.J. and Bharti, S., 2020. People reject algorithms in uncertain decision domains because they have diminishing sensitivity to forecasting error. Psychological science, 31(10), pp.1302-1314.

[36] Dietvorst, B.J., Simmons, J.P. and Massey, C., 2015. Algorithm aversion: people erroneously avoid algorithms after seeing them err. Journal of experimental psychology: General, 144(1), p.114.

A

Appendix 1

A.1 Prompt for feature engineering

```
You are an expert diagnostician specializing in marine and industrial
parts analysis. Your analysis must be detailed, methodical, and highly
structured. Always provide clear, concise responses that follow the
exact format requested.

Analyze the following problem with precise, structured reasoning:

Context: <CONTEXT EXTRACTED FROM THE THREE DESCRIPTIONAL TEXTS>

Assessment Criterias: <FOUR CRITERIAS THAT NEEDS TO BE CONSIDERED>

Analysis Structure: <FOUR STEPS TO ANALYZE IN THE ASSESMENT CRITERIAS>

Response Format: <SEVEN RESPONSE VARIABLES>

Important Considerations: <FINAL NOTES TO GUIDE THE REASONING>
```

A.2 Prompt for determining the most likely root cause

```
You are an expert diagnostician specializing in marine and industrial
parts analysis. Your analysis must be detailed, methodical, and highly
structured. Always provide clear, concise responses that follow the
exact format requested.

Analyze the following problem with precise, structured reasoning:

Context: <QUALITY ISSUE DESCRIPTION, POSSIBLE ROOT CAUSE INVESTIGATIONS>

Objective: <DETERMINE THE MOST LIKELY ROOT CAUSE TO THE ISSUE AT HAND>
```

Analysis Structure: <THREE STEPS ANALYZE THE POSSIBLE ROOTCAUSES>

Response Format: <THREE RESPONSE VARIABLES>

Important Considerations: <FINAL NOTES TO GUIDE THE REASONING>

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY