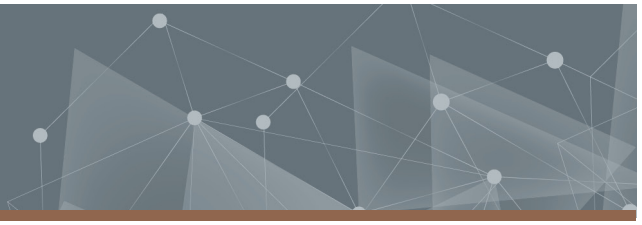
CHALMERS
UNIVERSITY OF TECHNOLOGY



Design and Evaluation of a RAG Chatbot in an Industrial Setting

Development of a Practical AI Prototype for Enhancing Knowledge Retrieval

Master's thesis in Complex Adaptive Systems

RASMUS NILSSON, OSCAR ROSMAN

DEPARTMENT OF PHYSICS

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

Master's thesis 2025

Design and Evaluation of a RAG Chatbot in an Industrial Setting

Development of a Practical AI Prototype for Enhancing Knowledge Retrieval

RASMUS NILSSON, OSCAR ROSMAN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2025

Design and Evaluation of a RAG Chatbot in an Industrial Setting
Development of a Practical AI Prototype for Enhancing Knowledge Retrieval
Rasmus Nilsson, Oscar Rosman

Supervisor: Martin Claesson, Volvo Group Trucks Technology
Examiner: Giovanni Volpe, Department of Physics

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

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Design and Evaluation of a RAG Chatbot in an Industrial Setting
Development of a Practical AI Prototype for Enhancing Knowledge Retrieval
Rasmus Nilsson, Oscar Rosman
Department of Physics
Chalmers University of Technology

Abstract

The rapid advancements in large language models (LLMs) have sparked industrial interest in leveraging generative AI for information access and decision support. However, the limitations of standalone LLMs, such as hallucinations and a lack of domain-specific or up-to-date knowledge, pose significant barriers in enterprise settings. Retrieval-Augmented Generation (RAG) architectures offer a promising solution by integrating external document retrieval into the generative process.
This thesis presents the design, implementation, and evaluation of a prototype RAG chatbot developed in collaboration with Volvo Group Trucks Technology (GTT). The chatbot enables users to interact conversationally with internal data comprising multiple modalities, including text and images. The system architecture was built from the ground up, focusing on document parsing, vector-based retrieval using FAISS, and multimodal integration. Evaluation covered functionality, generation quality, and usability, with positive feedback from end users supporting the system's practical viability.
This work contributes to applied research on RAG systems in industrial environments, highlighting challenges related to multimodal data handling, data security, and ethical considerations such as GDPR compliance. It serves both as a proof of concept for internal use and as a blueprint for broader AI adoption within the organization and similar industrial contexts.

Keywords: Retrieval-Augmented Generation (RAG), Large Language Models (LLMs), Multimodal Information Access, Knowledge Retrieval, Enterprise AI, Natural Language processing

Acknowledgements

We would like to start by thanking our supervisors at Volvo GTT, Martin Claesson, Thea Dencker and Jens Nordberg for their engagement and help in the development of the final product of this thesis. We would also like to thank our opponent Viktor Örnbratt for engaging conversations and good questions. We would also like to thank our contact person at the AI Core Team at Volvo GTT for guiding us through the systems and helping out when new obstacles were found. Thank you very much.

Rasmus Nilsson & Oscar Rosman, Gothenburg, June 2025

List of Acronyms

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

AI	Artificial Intelligence
FAISS	Facebook AI Similarity Search
GDPR	General Data Protection Regulation
GPT	Generative Pre-trained Transformer
LLM	Large Language Model
NLP	Natural language processing
RAG	Retrieval-Augmented Generator

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

Recent advancements in artificial intelligence (AI), particularly in the field of natural language processing (NLP), have sparked significant interest across industries. This momentum has been accelerated by the development of powerful large language models (LLMs) by companies such as OpenAI, Google, Anthropic, and Meta. These models have demonstrated unprecedented capabilities in understanding, generating, and processing human language, primarily in textual form, but increasingly also in modalities such as image, video, and audio. As a result, they have enabled a wide array of new applications across diverse sectors.

Enterprises are now actively exploring how these innovations can be leveraged to improve operational efficiency and gain a competitive edge. Potential use cases include automating repetitive tasks, augmenting employee workflows, enhancing decision-making, and redefining how organizations interact with customers and internal data.

Despite their strengths, LLMs also exhibit notable limitations. One of the most pressing issues is their tendency to hallucinate, producing responses that are fluent and plausible but factually incorrect. Furthermore, since these models are trained on static datasets, they lack access to real-time or domain-specific knowledge. This results in outdated or generalized answers when accurate, up-to-date, and context-specific information is needed, posing a significant barrier to adoption in environments where reliability and domain expertise are critical.

To address these limitations, a class of hybrid systems known as Retrieval-Augmented Generation (RAG) has emerged. RAG architectures combine the generative strengths of LLMs with the ability to query an external knowledge base. This grounding mechanism allows the LLM to incorporate relevant and current information into its responses, thereby reducing hallucinations and improving domain specificity. A further advantage is the flexibility of the knowledge base is that the information can be updated by simply adding or removing documents, eliminating the need for costly and time-consuming model retraining. These features make RAG systems particularly attractive for enterprise applications that demand accuracy, transparency, and adaptability.

This thesis was conducted in collaboration with a department focused on customer insight and solutions at Volvo Group Trucks Technology (GTT), which has initiated efforts to explore the potential of AI solutions in its operations. One area of interest is the democratization of information, enabling employees to access and interact with internal knowledge more easily and intuitively. Recognizing the promise of

RAG systems, the department proposed the development of a functional chatbot as a proof of concept. The goal was to evaluate the feasibility and impact of such a system before making broader investments in AI infrastructure.

1.2 Aim and Research Questions

The aim of this thesis is to design and develop a RAG chatbot that improves access to unstructured and multimodal information within an industrial setting. The focus is on building a flexible and functional system tailored to the needs of end users in the department at Volvo GTT. This applied research project investigates the following practical question.

How can a Retrieval-Augmented Generation chatbot be developed and evaluated to improve access to unstructured information for end users in an industrial setting like Volvo GTT?

1.3 Methodology Overview

This thesis employs an applied research methodology, focused on solving a practical problem through using existing technologies.

The project began with a brief literature review to establish a theoretical foundation and explore relevant frameworks. This was followed by a user-centered development process, where iterative feedback from supervisors and potential end users was gathered through weekly meetings. The system was built from the ground up, incorporating components for parsing diverse media types, retrieving relevant data, and generating contextual answers using an LLM.

Evaluation focused on the system's functionality and its ability to meet end-user needs. Both manual and automated methods were used to assess the quality of generated answers in relation to the retrieved context. The ethical implications of using generative AI in an industrial environment were also considered, especially in relation to data sensitivity and user trust.

1.3.1 Use of AI Tools

AI tools, including LLMs, were used during both the development and documentation phases of this project. During development, generative AI assisted in a limited capacity with code generation, debugging, and architectural planning of the RAG system. In writing this thesis, AI-based text assistants were occasionally used to rephrase, polish, or clarify passages, while maintaining strict human oversight and editorial control.

The intent behind these uses was to increase productivity and precision, not to delegate critical thinking or original contributions. All design decisions, analytical reasoning, and final edits remain the sole responsibility of the authors.

1.4 Scope and Limitations

This thesis focused on the design and development of a RAG chatbot tailored for a department working with customer insights and solutions at Volvo Group. The practical nature of the project, combined with organizational constraints and the limited timeframe of a master's thesis, shaped both the scope and limitations. The goal was to deliver a proof of concept and prototype that enables users to interact conversationally with internal data for the first time.

1.4.1 Scope

The scope of the thesis was defined by the practical aim of developing a working system rather than conducting theoretical research. The main aspects covered include:

1. **Architecture:** A modular RAG system was implemented, with the primary focus on the retriever component. The generator component received limited attention, mostly involving prompt engineering and parameter tuning.
2. **Application:** Although interface design was not the core research focus, a functional Graphical User Interface (GUI) was developed to enable users to interact with the system via natural language conversation.
3. **Data modalities:** The system was designed to handle multiple data formats. However, most documents were ultimately converted into two primary modalities, text and images, to simplify retriever and parsing pipelines.
4. **Evaluation:** Evaluation was centered on two aspects: (1) the effectiveness and efficiency of information retrieval, and (2) the usability and user experience of the application interface.
5. **Expectations:** The Volvo GTT department viewed the project as a prototype and proof of concept. Accordingly, the focus was on exploring features and ensuring core functionality rather than delivering a fully polished, production-grade application.

1.4.2 Limitations

Several limitations influenced the outcomes of the project:

1. **Data access:** Due to the sensitive nature of internal data, all project members signed non-disclosure agreements. While this had minimal impact on development, it limits what can be shared in this thesis. In some instances, data was anonymized or omitted entirely.
2. **Organizational access:** The size and structure of Volvo GTT presented challenges in accessing certain systems, tools, or information, which in turn imposed constraints on system design and integration.
3. **Evaluation constraints:** Time limitations restricted the depth of evaluation. While valuable feedback was gathered through weekly meetings with end users and supervisors, no formal long-term studies or comprehensive user testing could be conducted.

4. **Generality:** The solution was developed specifically for Volvo GTT's context, using tools and data available within the organization. While some components, like document parsing and retrieval, were designed for reusability, the system was not tested or validated in other industrial domains or with alternative use cases.

1.5 Contribution and Significance

This section outlines the significant impact of this thesis, detailing its contributions to both academic discourse and practical industrial applications, while also identifying key areas for future exploration. It highlights the development of a unique multimodal RAG system, its practical benefits for Volvo GTT, and how it lays the groundwork for further advancements in AI-powered knowledge management.

1.5.1 Academic Contribution

While the individual components and methods used to construct the RAG chatbot are previously established, this thesis's academic contribution lies in the holistic design and evaluation of an integrated multimodal RAG system for spatially complex and non-standardized documents. A use case that remains underexplored in academic literature.

1.5.2 Practical Contribution

This thesis represents a practical exploration of AI implementation within a department at Volvo GTT. The resulting prototype is a functional RAG chatbot that enables end users to query and interact with unstructured internal data using natural language. The system serves both as a proof of concept and as a potential foundation for future internal tools. More broadly, the project offers a blueprint for how industrial organizations can approach the adoption and deployment of AI-powered knowledge access systems based on RAG methodologies.

1.5.3 Future Research

While this thesis presents a working prototype and demonstrates feasibility, several directions remain open for future research and development. These range from improved functionality to deeper evaluation of the product.

1. **Long-term Evaluation:** Future studies could incorporate structured, in-depth assessments of user satisfaction, retrieval accuracy, and the system's impact on productivity or decision-making.
2. **Multimodal Retrieval:** Further research is needed to improve how RAG systems integrate and retrieve information from diverse modalities, including audio and video. Enhancements in the parsing, storage, and retrieval of complex formats such as diagrams, tables, annotated images, and handwritten notes would also be beneficial.

3. **Automated Chunking of Visual Layouts:** More advanced techniques for the automated understanding and chunking of spatially distributed content, such as that found on whiteboards or in mind maps, could significantly enhance the system's ability to handle non-linear or non-textual documents.

1.6 Structure of the Thesis

Chapter 2: Theoretical Background lays the groundwork by explaining essential concepts related to LLMs, the architecture of RAG systems, and various retrieval mechanisms. Building upon this, Chapter 3: Modeling of Product delves into the specific design and architecture of the developed RAG chatbot.
The methodological approach is detailed in Chapter 4: Methods, covering the dataset used, the chosen evaluation metrics, and the testing strategy. Subsequently, Chapter 5: Results presents the findings derived from the evaluation process. These results are then thoroughly discussed in Chapter 6: Discussion, which also addresses the limitations of the study and suggests avenues for future work. Finally, Chapter 7: Conclusion summarizes the key insights gained and the challenges encountered throughout the research. The thesis concludes with an Appendix, providing supplementary information.

2
Theoretical Background

2.1 Large Language Models

LLMs are a part NLP, which in turn is a subfield to AI. Natural language generation (NLG) focuses on producing coherent and contextually coherent text. The field was introduced in 1964 by the rule-based models such as ELIZA [1] and has evolved into today's models based on Generative Pre-trained transformers (GPTs) which has improved contextual understanding and versatility of the field.

2.1.1 Fundamentals of text generation

Tokenization

Tokenization is the initial step in processing input for a language model, where the raw text, typically a string of characters is segmented into smaller units known as tokens. This transformation enables the model to interpret and operate on the input in a structured manner. For a human, tokenization can be thought of as dividing a sentence or document into smaller linguistic components, such as words or sentences. In the context of LLMs, these tokens often correspond to individual words or subword units [2].

Encoding and Embedding

Modern language models represent language in a high-dimensional vector space, where each token is encoded as a continuous vector. These vector representations, known as embeddings capture semantic and syntactic relationships between tokens, enabling the model to reason about language in a structured way.
Earlier approaches to representing tokens relied on one-hot encoding, where each token corresponds to a high-dimensional sparse vector with a single non-zero entry. While simple, this method fails to capture any notion of similarity between different tokens.
In contrast, modern models assign each token a unique integer ID based on a predefined vocabulary. These IDs are then mapped to dense, low-dimensional vectors using a learned embedding matrix. The values in this matrix are optimized during training so that tokens used in similar contexts end up with similar representations. This process allows the model to internalize meaningful relationships between words, phrases, and even entire sequences, giving the model a semantic understanding and context [3].

Autoregressive Text Generation

Language models generate text by predicting the next token in a sequence based on the tokens that precede it. Rather than directly responding to a prompt in a goal directed way, the model continues the input sequence it receives by extending it one token at a time. The generation process is governed by the chain rule of probability, as shown in Equation 2.1, where each token x_i is conditioned on all previous tokens $x_1, \ldots, x_{i-1}$. Variations of equation 2.1 are common, particularly using log-probabilities instead to ensure numerical stability.

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | x_1, x_2, \ldots, x_{i-1}) \tag{2.1}$$

Once the probabilities have been calculated, the model could select the most probable but this has been observed to produce a deterministic, rigid text close to the data the model has been trained on. Instead different sampling strategies, which are discussed in section 2.1.1 are employed to generate more natural answers for the conversation. After the next token has been selected and added to the sequence the model relies on regression, treating the new sequence as input to predict the next token until a token limit is reached or a stop token is generated marking the end of the generated text [4].

Sampling Strategies

Sampling strategies, which determines how the next token is selected based on the model's predicted probability distribution, have significant impact on shaping the behavior of generative models and the quality of the generated text. The strategies can prevent overly deterministic responses that closely mirror the training data, guide the tone of the output, and help the model handle unusual or ambiguous input sequences more effectively.
While all sampling methods rely on the model's computed probability distribution over possible next tokens, they introduce stochasticity into the selection process to varying degrees. These methods differ in how they modify or constrain the distribution from which the next token is sampled, influencing the creativity, coherence, and variability of the text. The two most commonly used strategies are temperature scaling and nucleus (top-p) sampling. Temperature adjusts the sharpness of the probability distribution. A high temperature, close to 1, flattens the distribution, making lower-probability tokens more likely and thus increasing randomness. A low temperature, approaching 0, sharpens the distribution, concentrating probability mass around the most likely tokens and resulting in more deterministic, focused outputs. Nucleus sampling, or top-p sampling, takes a dynamic approach. Instead of fixing the number of candidate tokens, it selects the smallest subset of tokens whose cumulative probability exceeds a predefined threshold p. This allows the model to flexibly adjust the size of the sampling pool depending on the certainty of the context, retaining diversity when the model is unsure, and narrowing focus when it is confident.
Each strategy can significantly influence the generated text and serves as a low-cost method to adjust a generative model's behavior without retraining or modifying

the underlying data. In practice, these strategies are often used in combination, for example, applying nucleus sampling together with temperature scaling, to fine-tune the output characteristics and better align the language model's behavior with the desired application [5].

2.1.2 Transformer Architecture

The most capable language models in use today are based on the transformer architecture, originally introduced by Vaswani et al. in 2017 with the revolutionizing paper *Attention Is All You Need* [6]. The innovation of the transformer architecture introduced new capabilities shifting the focus of the NLP field from the previously popular recurrent neural networks (RNN), long-short-term memory (LSTM) and convolutional neural networks (CNN) models. Previous models struggled with long-range dependencies due to vanishing and exploding gradients and were bound to sequential computations making them inefficient. The transformer architecture relying on attention mechanisms, proved to handle long-range dependencies well and could parallelize parts of the computation, which made them more efficient while improving the performance.
The original transformer architecture, depicted in Figure 2.1, comprises two main components: an encoder and a decoder. These components are typically used together in sequence-to-sequence tasks, such as machine translation, where the encoder processes the input sequence and the decoder generates the output. However, many modern LLMs, particularly those focused on text generation like the GPT models, primarily use a decoder-only architecture [4], [7].

Attention Mechanism

The attention mechanism allows the transformer model to dynamically weigh the importance of the input sequence when processing the sequence. Unlike traditional recurrent networks that process information sequentially, attention enables direct connections between any two positions in a sequence, regardless of their distance. This capability is what allows it to detect the long-range dependencies effectively.
At its core, the attention mechanism calculates a weighted sum of input values, where the weights or attention scores are determined by the relevance between a query and a set of keys. Specifically, for each element in a sequence, the mechanism compares it against all other elements to compute a compatibility score. These scores are then normalized, typically using a softmax function, to produce a probability distribution that dictates how much attention should be paid to each corresponding value. The output of the attention mechanism is the sum of these values, weighted by their attention scores.
The Transformer architecture primarily leverages two forms of attention, as illustrated in Figure 2.2:

1. **Self-Attention:** This mechanism allows the model to relate different positions of a single sequence to compute a representation of that same sequence. For instance, when processing a word in a sentence, self-attention enables the model to consider how other words in that same sentence contribute to its

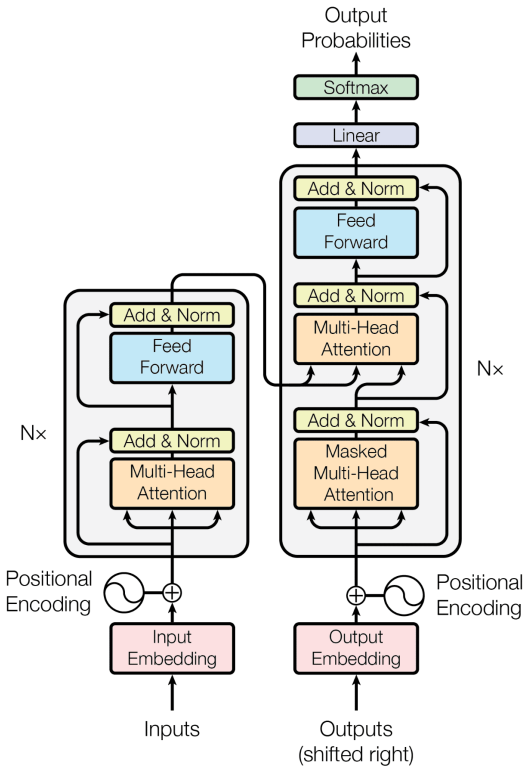


Figure 2.1: The Transformer architecture as presented by Vaswani et al. [6]. It consists of an encoder (left) and a decoder (right) stack, primarily relying on multi-head attention mechanisms.

meaning. This is achieved by having queries, keys, and values all derived from the same input sequence.

2. **Multi-Head Attention:** Instead of performing a single attention function, Multi-Head Attention linearly projects the queries, keys, and values multiple times with different, learned linear projections. It then performs attention in parallel for each of these projected versions. The outputs from these attention heads are then concatenated and once again linearly transformed to produce the final result. This multi-headed approach allows the model to jointly attend to information from different representation subspaces at different positions, improving its understanding.

Encoder

The encoder component of the Transformer architecture is responsible for processing the input sequence and transforming it into a contextualized representation. It comprises a stack of identical layers, typically six in the original design. Each layer consists of two primary sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network [6].
The input sequence first undergoes an embedding process, where each token is converted into a dense vector representation. Crucially, as the Transformer lacks recurrence, positional encodings are added to these embeddings. These encodings inject

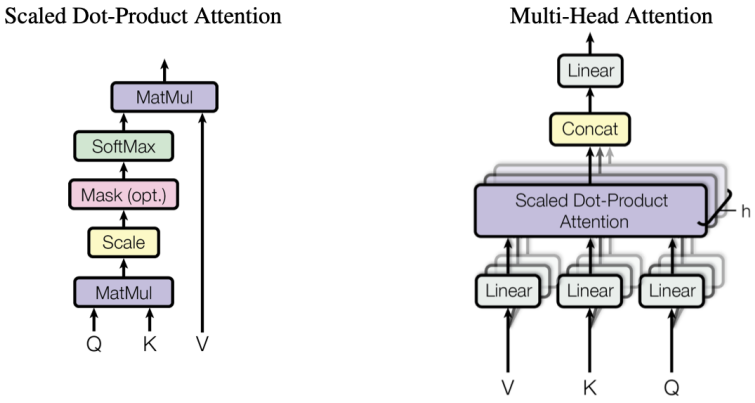


Figure 2.2: Visual representation of Scaled Dot-Product Attention (left) and Multi-Head Attention (right), adapted from Vaswani et al. [6].

information about the relative or absolute position of tokens in the sequence, allowing the model to distinguish between tokens that appear in different positions but have the same meaning [8]. The output of the embedding layer, combined with positional encodings, is then passed through the stacked encoder layers. Within each encoder layer:

1. The first sub-layer is a Multi-Head Self-Attention mechanism. This allows the encoder to weigh the importance of all other tokens in the input sequence when processing a specific token, thereby capturing long-range dependencies and contextual relationships within the input itself. A residual connection is employed around this sub-layer, followed by layer normalization.
2. The second sub-layer is a simple, position-wise fully connected feed-forward network. This network is applied independently and identically to each position in the sequence, further transforming the representations learned by the attention mechanism. Again, a residual connection and layer normalization are applied.

The output of the final encoder layer is a sequence of contextualized vector representations, keys and values that summarize the input sequence, which are then passed to the decoder for generating the output.

Decoder

The decoder component is designed to generate the output sequence, typically one token at a time, in an autoregressive manner. In its full form, it utilizes both the previously generated output tokens and the contextualized representations produced by the encoder. Like the encoder, the decoder consists of a stack of identical layers, usually six in the original Transformer model [6].
Each decoder layer contains three primary sub-layers:

1. The first sub-layer is a masked Multi-Head Self-Attention mechanism. This masking is crucial for autoregressive decoding: it prevents positions from attending to subsequent positions. This ensures that the prediction for a given

token t can only depend on the known outputs at positions less than t, mimicking sequential generation. As in the encoder, a residual connection and layer normalization follow.

2. The second sub-layer is a Multi-Head Attention mechanism that attends to the output of the encoder stack. This is often referred to as "encoder-decoder attention" or "cross-attention." Here, the queries come from the previous decoder sub-layer, while the keys and values are derived from the output of the encoder. This mechanism allows the decoder to focus on relevant parts of the input sequence when generating each output token, facilitating the transfer of information from the source context. Residual connection and layer normalization are again applied.
3. The third sub-layer is a position-wise fully connected feed-forward network, identical in structure to that in the encoder. It further processes the combined representations from the attention mechanisms. Residual connection and layer normalization complete this sub-layer.

The output of the final decoder layer is then projected through a linear layer and a softmax function to produce probabilities over the vocabulary, from which the next output token is selected. This iterative process allows the decoder to build the output sequence step by step, leveraging both its generated history and the input's context.

2.1.3 Prompting

Prompting is an increasingly important technique for influencing the behavior of LLMs without modifying their underlying parameters. The term prompt refers to the input text provided to the model, which can significantly affect the nature and quality of the generated output [9], [10].

Prompting has several key applications. One use is in system prompts, which are employed by organizations such as OpenAI, Anthropic, and Google to guide the default behavior of their deployed LLMs. These hidden prompts help shape the model's tone, politeness, response style, and adherence to ethical boundaries [11].

Another major application is in enhancing task performance. Zero-shot and few-shot prompting are paradigms where the model is given no, or only a few examples of the task within the prompt itself. These strategies have been shown to improve performance on a variety of NLP tasks without task-specific fine-tuning [9].

Moreover, prompting can influence not just what the model says, but how it reasons. Chain-of-thought prompting is a technique in which examples of intermediate reasoning steps are included in the prompt. This has been shown to induce multi-step reasoning capabilities in LLMs and significantly improve performance on arithmetic and commonsense reasoning tasks [10].

Overall, prompting serves as a versatile interface for steering LLMs in both task execution and behavioral alignment.

2.1.4 Limitations of LLMs

While LLMs have demonstrated significant capabilities in generating human-like and coherent text, they still possess several important limitations that need to be considered.

Information Accuracy

A key challenge with LLMs is hallucinations. This occurs when the LLM generates incorrect or made-up information and presents it confidently as facts [12]. This arises from the predictive nature of LLMs, which rely on statistical patterns in its training data rather than access to verified external sources [12].
LLMs also suffer from a knowledge cutoff, meaning they cannot access events or information beyond the time of their training [13]. Updating this internal knowledge requires retraining or fine-tuning, which is both computationally expensive and time-intensive [14]. In addition, since information within an LLM is stored implicitly in model parameters, it is not directly traceable to its original source, complicating fact-checking and verification [15].

Context Length

While transformer-based LLMs can process longer inputs than earlier models, they are still constrained by a maximum context length. Providing very long inputs, such as entire documents, can lead to degraded performance, as models may overlook or misprioritize important information in the middle of the input [16]. This can limit the effectiveness of LLMs in tasks that require reasoning over large or complex input sequences.

2.2 Retrieval-Augmented Generation

RAG is an approach that integrates information retrieval and language generation to create an up-to-date and domain-specific language model that lets the user control sources and verify the information's correctness. Shown below in Figure 2.3 is a basic version of what a RAG is.

2.2.1 Motivation for RAG

RAG directly addresses the critical limitations of an LLM by combining the generative power of LLMs with the precision and correctness of information retrieval systems. The core motivation behind RAG is to ground LLM generations in verifiable, external knowledge, thereby significantly enhancing factual accuracy, contextual relevance, and explainability.
RAG dynamically retrieves relevant, up-to-date information from external knowledge bases (e.g., document repositories, databases) and provides this retrieved context to the LLM prior to generation. This process effectively grounds the model's output in verifiable facts, thereby substantially mitigating the risk of hallucination [17], [18]. Lewis et al.'s foundational work [17] demonstrated RAG's effectiveness in

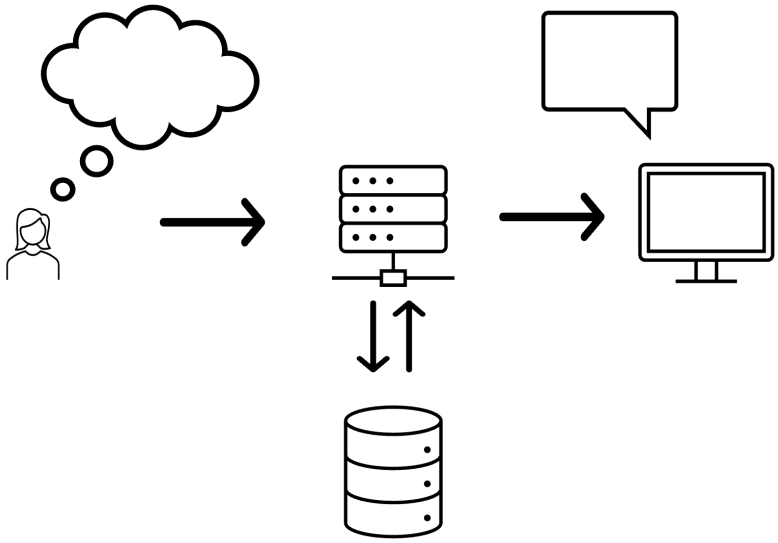


Figure 2.3: Basic version of RAG where the user sends a query to the program that uses it as a search in the database to retrieve information that is sent to the LLM and used to create an answer back to the user.

knowledge-intensive tasks by integrating a non-parametric memory (retrieval component) with a parametric memory (generation component), allowing the model to leverage a much broader and continually updated knowledge source.
By supplying specific, relevant information, RAG enables the LLM to generate responses that are not only factually accurate but also highly tailored and contextually relevant to the user's query [18]. This external grounding also inherently improves the explainability and trustworthiness of the generated content. Users can trace the information back to its original source, fostering greater confidence in AI-generated outputs and facilitating audit ability [19], [20].
Crucially, RAG offers a cost-effective and dynamically adaptable alternative to continuous LLM fine-tuning. While fine-tuning embeds knowledge statically and requires expensive retraining as data evolves, RAG allows LLMs to access new information without modifying the underlying model parameters [21], [22]. This separation of knowledge from the model's weights makes RAG a more agile and economically viable solution for dynamic knowledge environments, reducing the prohibitive costs associated with frequent LLM retraining [21], [22].

2.2.2 Knowledge Base

The efficacy of RAG fundamentally relies on the existence and characteristics of an external knowledge base [17]. This component serves as the main, dynamic repository of factual information that augments the internal, parametric knowledge of LLMs. Unlike the static, pre-trained knowledge embedded within an LLM's weights, the knowledge base provides a dynamic and extensible source of verifiable facts, enabling RAG systems to overcome limitations such as knowledge cutoffs and the generation of hallucinated content [23], [24]. The factual foundation of a RAG

system is directly proportional to the quality, comprehensiveness, and structure of its associated knowledge base.

Characteristics of a RAG Knowledge Base

A robust knowledge base for RAG exhibits several key factual characteristics that contribute to its utility:
Verifiability and Grounding: The primary function of the knowledge base is to provide concrete, verifiable information to the LLM. Each piece of information within the base should ideally be traceable to its original source, allowing for a strong factual grounding of the generated responses [25]. This characteristic is crucial for building trust and ensuring the reliability of RAG outputs, especially in high-stakes applications.
Currency and Updatability: Unlike the inherent knowledge cutoff of LLMs, a well-designed knowledge base can be continuously updated with new information. This dynamic nature ensures that the RAG system can access and utilize the most current facts, making it suitable for rapidly evolving domains such as current events, scientific discoveries, or frequently revised legal and medical guidelines [24].
Domain Specificity and Breadth: The knowledge base is typically tailored to the specific domain of application. This allows for the inclusion of highly specialized or niche information that general-purpose LLMs lack. While LLMs excel at broad conversational abilities, the knowledge base provides the necessary depth and precision for expert-level interactions within a defined scope [26]. Its breadth, in terms of encompassing all relevant information for the domain, directly impacts the completeness of RAG responses.
Semantic Accessibility: The factual content within the knowledge base is represented in a manner that facilitates semantic search. Through methods like dense vector embeddings, the semantic meaning of information chunks is captured, allowing the retrieval component to identify facts conceptually related to a user's query, even if specific keywords are not present [27], [28]. This enables the system to retrieve relevant context that might not be discoverable through simple keyword matching.
Scalability: For real-world applications, the knowledge base must be capable of storing and efficiently retrieving from vast quantities of information, potentially spanning millions or billions of documents or facts. This necessitates scalable indexing and storage solutions that can handle large datasets without compromising retrieval latency [28].
In essence, the knowledge base serves as the externalized memory of the RAG system, directly influencing its ability to generate accurate, relevant, and trustworthy outputs. Its factual integrity and accessible structure are crucial for RAG's practical deployment in knowledge-intensive applications.

Structured and Unstructured Knowledge base

The retrieval from the knowledge base is dependent on how the data has been structured inside it. A comprehensive knowledge base often integrates both unstructured textual data and structured data. Unstructured data refers to data such as documents and articles while structured data refers to data such as databases

and knowledge graphs [29]. By incorporating metadata and defined labels into raw data, previously unnoticed connections and relationships within the dataset can be identified and leveraged to enhance the relevance of retrieved information [25]. This augmentation provides richer contextual signals for the retrieval mechanism, extending beyond simple keyword matching or dense vector similarity. Furthermore, the construction of a knowledge base explicitly based on structured data, such as knowledge graphs, has been shown make use of deeper relational patterns inherent in the data [29]. Such structured representations are capable of providing more precise and granular context to an LLM compared to knowledge bases relying solely on raw, unstructured data, thereby improving the factual grounding and reasoning capabilities of RAG systems [26], [29].

2.2.3 Retriever

Following the construction and organization of the knowledge base, the retriever component constitutes the next critical stage in a RAG system. Its primary function is to efficiently identify and extract the most relevant information from the expansive knowledge base in response to a given user query [17], [23]. The effectiveness of the retriever directly dictates the quality of the context provided to the LLM, thereby influencing the factual accuracy and relevance of the generated response.

The retrieval process typically commences by transforming the user's natural language query into a machine-understandable representation, most commonly a high-dimensional vector embedding. This is achieved using specialized encoding models, such as bi-encoders (e.g., Sentence-BERT [27]), which map both queries and knowledge base documents (or their chunks) into a shared semantic space. Once embedded, the query vector is used to perform a similarity search within the indexed vector space of the knowledge base. This search identifies documents or passages whose embeddings are semantically closest to the query embedding, indicating their potential relevance [28].

While initial retrieval often prioritizes semantic similarity (dense retrieval), hybrid approaches may also incorporate keyword-based matching (sparse retrieval) to capture keyword overlap, especially for highly specific queries [24]. Furthermore, to refine the set of initially retrieved documents and present the most relevant information to the LLM, a re-ranking stage is frequently employed. Re-rankers, such as cross-encoders, take both the query and each retrieved document as input and score their relevance more accurately than the initial retriever, effectively filtering out less relevant noise and prioritizing truly valuable context [25]. Also, LLMs have been known to be used as a judge to determine the relevance of the retrieved data. This due to its ability to see context and not only semantic similarity [30]. This multi-stage retrieval process ensures that the LLM receives a concise and highly relevant set of factual inputs, minimizing the risk of contextual dilution and supporting robust, grounded generation.

2.2.4 Generator in RAG

Following the retrieval of relevant information from the knowledge base, the retrieved documents or passages are passed to the generator component, which is typically a pre-trained LLM. In the RAG framework, the LLM's role shifts from generating responses based solely on its internal parametric knowledge to synthesizing answers that are explicitly informed and grounded by the provided external context [17].
The generator receives the original user query alongside the ranked and filtered relevant information from the retriever. Its primary task is to integrate this external facts with its natural language generation capabilities to formulate a contextually relevant, and factually accurate response. This process fundamentally transforms the LLM's behavior, directing its generation towards verifiable information rather than relying purely on learned patterns that can lead to hallucinations or outdated content [23], [24].
The benefits of augmenting an LLM with a retrieval mechanism are many for the generator. Firstly, by providing concrete evidence, RAG significantly limits the LLM's relevance to generate inaccurate or fabricated information, ensuring that responses are anchored in reliable sources [18]. Secondly, the generator gains access to information beyond its training cutoff, including the most current events and specialized domain knowledge, enabling it to provide relevant answers for dynamic or niche fields [24]. Thirdly, the ability to trace generated facts back to their source within the retrieved documents improves the transparency of the LLM's reasoning, fostering greater user trust and facilitating auditability [31]. Lastly, with targeted relevant documents, the generator can produce more precise and nuanced answers that directly address the specifics of the user's query and the nuances of the retrieved context.
While the retriever provides the necessary factual input, the generator remains crucial for its ability to understand complex queries, synthesize information across multiple retrieved documents, and articulate sophisticated responses in natural language. The interplay between these two components defines the core functionality of a RAG system, aiming to achieve a balance between fluency, accuracy, and comprehensiveness.

2.3 Data Management

This section outlines the critical aspects of data management related to the development and deployment of RAG chatbots within an enterprise environment. It addresses both the regulatory landscape governing data handling and the inherent data-related risks associated with LLMs.

2.3.1 General Data Protection Regulation

GDPR (Regulation (EU) 2016/679) is a European Union law governing data protection and privacy, with broad extraterritorial reach, making it critical for global enterprises like Volvo GTT [32]. Its core objective is to empower individuals with control over their personal data. For a RAG chatbot, key GDPR principles that de-

mand attention include: *Lawfulness, Fairness, and Transparency* in data processing; *Purpose Limitation*, ensuring data is used only for its intended function; *Data Minimization*, advocating for the collection of only necessary data; *Accuracy* of personal data; *Storage Limitation*, retaining data no longer than necessary; and critically, *Integrity and Confidentiality (Security)*, mandating robust measures against unauthorized access or loss [33]. Adhering to these principles, alongside respecting data subject rights (e.g., right to erasure), necessitates careful data governance, including anonymization and strict access controls, to ensure compliance in enterprise RAG deployments [34].

2.3.2 LLM Concerns

Deploying LLMs and RAG chatbots in an enterprise environment introduces significant data leakage risks, defined as the unintended disclosure of sensitive information through the system's interactions [35]. This risk is particularly relevant when processing confidential company data.
Key mechanisms leading to data leakage include:

1. **Prompt Injection and Jailbreaking**: Malicious inputs can manipulate the LLM to reveal sensitive data or bypass safety mechanisms, including information retrieved by the RAG system [36].
2. **Over-retrieval and Context Window Exploitation**: The retriever might fetch sensitive but irrelevant data, which, if passed to the LLM's context window, could be inadvertently exposed by subsequent prompts [37].
3. **Lack of Granular Access Control**: Insufficient access restrictions within the knowledge base, unapplied during retrieval, can lead to unauthorized information disclosure.

Mitigation strategies are crucial for a secure RAG deployment. These include data anonymization/pseudonymization before processing, robust access control mechanisms for the knowledge base, careful prompt engineering, and implementing output filtering layers to prevent sensitive information from reaching the user [11], [38].

3
Modeling of Product

This chapter lays out the architecture of the final product. The various components and pipelines are detailed and explained to provide insight into the system's design. Finally, the challenges encountered during the project are listed, and the decisions made to avoid obstacles are discussed.

3.1 Parsing of Media

The data stored in the knowledge base origins from .pptx-, ppt- and pdf-files. File types like .doc and .docx have been excluded due to technical parsing limitations. Other media types, such as .xlsx and .mp4 files, were excluded due to their minimal presence in the dataset.
A pipeline was developed to extract meaningful data and store it as structured text, ensuring efficient and accurate retrieval. Accuracy was prioritized over execution time during database creation, leading to the selection of rigorous and proven methods over faster but less reliable alternatives. The data mainly consisted of PDF and PowerPoint files that where created by different individuals. Extracting structured data from PowerPoint and PDF files required a robust pipeline due to the varied and non-standardized nature of the content. A more detailed discussion of these challenges is presented in section 3.5.
As illustrated in Figure 3.1, by running the main program, files are sorted into PDF and PowerPoint categories for processing by their respective parsers. This ensures an easy to use program for future updates.

3.1.1 PDF Parsing

PDF files are parsed using the fitz module from the Python library PyMuPDF, which extracts both text and images. The parsing is performed page by page, extracting all text and images. Extracted text from each page is stored in a .txt file, and images are saved in their original file format. The process of image extraction presents certain challenges, which are discussed further in Section 3.5.
Extracted text is split into chunks by an encoder based on a maximum token length. The encoder respects sentence boundaries and applies overlapping to minimize context loss and avoid breaking sentences. Each chunk is stored in a dictionary that includes relevant metadata. This metadata is later combined with that of the extracted images to form the final data structure presented in Figure 3.3.
Extracted images are filtered by common image extensions such as .png, .jpg, and

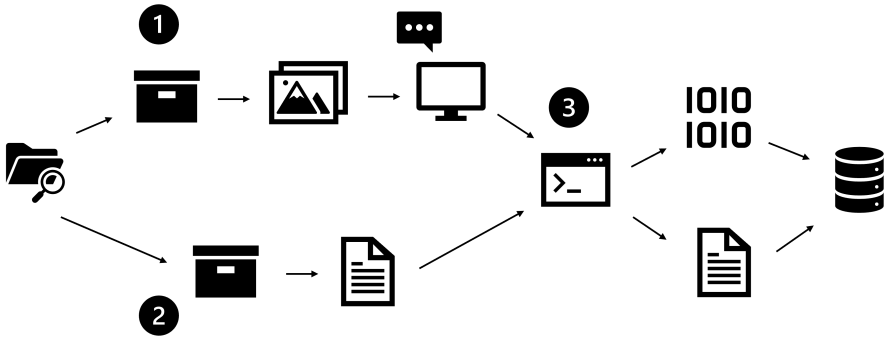


Figure 3.1: The flowchart explains how the data is retrieved from a folder and fed through a pipeline to end up as vectorized data with a correlating .json-file containing the metadata for each chunk. Path 1 shows how images are extracted from the file, stored and fed through the Mulitmodal language model to get a good description by using a prompt designed to describe vital information and extract keywords. Path 2 shows how text is parsed from .pdf-files to be stored in a .json-file. These two .json files are later combined in step 3 and the chunk of information is encoded, indexed and stored in a FAISS-database. The same indexed is put on the metadata file that is stored for retrieval.

.jpeg. To ensure compatibility with Chat GPT-4o, images are encoded using base64 and then processed by the model using a prompt carefully designed to maximize relevant information while minimizing noise. The outputs are stored in an image metadata dictionary and later merged with the text metadata to form the final structured format, as presented in Figure 3.3, which includes source references and user-relevant information.

3.1.2 PowerPoint Parsing

PowerPoint files posed a significant challenge in extracting both text and images from each slide. This is discussed further in Section 3.5. To handle the non-standardized nature of PowerPoint content from different creators, a general solution was required, one that could produce a concise but informative description of each slide, including relevant keywords and important concepts. After evaluating several options, OpenAI's GPT-4o was avalailable and selected for its multi modal capabilities. Other image captioning models, like BLIP, was used but could never create text complex enough to capture the important information from the slides.
The PowerPoint files are processed using the Python library comtypes, which opens each file in Microsoft PowerPoint and converts every slide into a .png image. These images are saved for further processing. Once all images from PowerPoints and PDFs are extracted, they are processed using the same procedure described in the final paragraph of Section 3.1.1.

3.1.3 Encoding and Storing Data

Encoding the data was done using the all-MiniLM-L6-v2 from the sentence_transformers library. This model encodes the text parts from the image description and text

chunks into 384 dimensional vectors. This dimension supports efficient and effective retrieval.
Facebook AI Similarity Search (FAISS) is used to create an index database for the vectors. The chunk of text is encoded and vectorized. Each vector is associated with its corresponding metadata through a shared index. The metadata is stored in a knowledge base, while the vectors are stored in an index database. This approach enables efficient and accurate vector retrieval. Given the nature of the original raw data, an unstructured database was chosen due to time limitations.

3.2 RAG Architecture

Within the implemented RAG architecture, the process of identifying and preparing relevant context for the LLM involves a multi-stage pipeline designed to enhance both efficiency and precision. This pipeline ensures that the LLM receives a curated set of highly pertinent information from the knowledge base.
The initial retrieval phase utilizes the FAISS library's search functionality. Given a user query, FAISS performs an Approximate Nearest Neighbor (ANN) search within the vector space of the knowledge base, efficiently identifying the 20 semantically closest document chunks. This initial retrieval exploits the dense vector embeddings to capture the conceptual similarity between the query and the knowledge base content.
Subsequently, to refine this initial set of 20 retrieved chunks, a re-ranking mechanism is employed. An all-MiniLM-L6-v2 cross-encoder model is used to score the pairwise relevance between the user query and each of the 20 retrieved chunks. Unlike bi-encoders used in initial retrieval, cross-encoders process the query and document pair jointly, allowing for a deeper contextual understanding and more accurate relevance assessment. Based on these re-ranking scores, the top 5 most relevant sources are selected to proceed to the next stage.
Finally, these top 5 sources undergo a relevance judgment by the LLM itself. The LLM is prompted to evaluate each source's relevance to the original user query, assigning a grade on a scale of 1 to 5. This human-like assessment provides a final layer of quality control. Only sources achieving a relevance grade of 3 or higher are then compiled into the final context provided to the LLM for generating the ultimate response to the user query. This multi-layered retrieval and filtering strategy aims to maximize the quality and factual grounding of the information presented to the generator.
The decision to extract 20 sources and feed the top 5 results from the cross-encoder into the "LLM-as-judge" part was made based on the evaluation of the results displayed in Chapter 5.

3.3 Interface Design and Implementation

Initially, the interface prototype was developed using Streamlit, a Python library that provided pre-built templates, which facilitated the rapid exploration of desired

features. This allowed for an efficient prototype phase of development, focusing on understanding the core functionalities needed for the application.
Following this exploration, Streamlit was replaced by NiceGUI. This transition was motivated by NiceGUI's suitability for more complex applications, offering enhanced low-level control over interface elements. NiceGUI also stood out due to its comprehensive documentation, its cross-platform compatibility and enabling development as a local application with the flexibility to deploy it as a web application with only minor code modifications.

3.3.1 Chat Window Functionality

The chat window is designed for user familiarity, mimicking the expected behavior of typical chatbots. Crucially, it supports Markdown formatting, a feature deemed essential after observing that the LLMs used in the project consistently preferred to generate responses in this format. Embracing Markdown allowed presenting neatly formatted responses, structured lists, and emphasizing key points, enhancing readability and user experience.
The core logic for sending and receiving messages is integrated within the chat window, as illustrated in Figure 3.2. When a user submits a query, the interface immediately updates to display the message. Subsequently, the system initiates the answer generation process. This involves several steps:

1. **Document Retrieval**: The system first identifies the most relevant documents for the query within the FAISS vector database.
2. **Cross-Encoder Selection**: A cross-encoder then refines this selection, choosing the most similar documents.
3. **LLM-based Filtering**: Finally, the reduced set of documents is further filtered based on relevance using the same LLM responsible for handling the conversation.

Once the relevant sources are identified and filtered, they are passed as context to the LLM, along with the user's query and a limited conversation history. If enabled, these selected sources are attached to the generated response, providing transparency and traceability. Otherwise, the LLM's response is sent directly to the interface.

3.3.2 Conversation history

The chatbot's interface incorporates a robust conversation history feature, allowing users to revisit all past interactions. Each conversation within the chat window is internally tracked as a list of dictionaries, with each dictionary containing "role", user or assistant and "content" keys. The order of these dictionaries in the list dictates the display sequence of messages, with content formatted according to its assigned role.
Conversation histories are persistently stored as JSON files, with each file named after the conversation and updated with its last modification date and time. New conversations are automatically titled "New conversation" and are appended with a

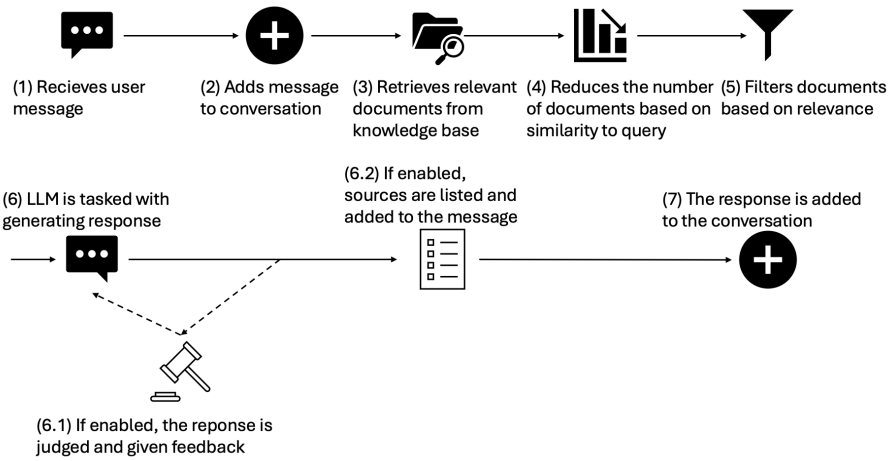


Figure 3.2: The figure is a visual representation of the logic within the chatbot from that a message from the user is received and that the chatbot has produced a response.

numerical suffix (e.g., "New conversation (1)", "New conversation (2)") if a conversation with the same name already exists. Users have full control over these stored conversations, with the ability to open, rename, and delete them independently.

3.3.3 Settings

The chatbot offers a comprehensive settings menu, providing users with greater control over its behavior than typically found in other chatbots. This customization caters to the different needs within the department.

Language Model Selection

Volvo GTT utilizes several LLMs, each possessing distinct characteristics. Since all these models are accessible via the OpenAI library and the same API key, the chatbot allows users to select their preferred LLM. This choice is particularly beneficial given that models may experience downtime due to maintenance or updates by Volvo's IT or AI departments. Furthermore, as models are occasionally retired or introduced, the application allows users to add new models to their options, though it remains restricted to chatbots available on the Volvo GTT network.

Choose Personality

An effective method to influence an LLM's behavior is through system instructions, which remain invisible to the user during conversation. To enhance the chatbot's relevance across various contexts, for example idea generation or fact-finding, users can select a predefined personality for the chatbot. The application comes pre-loaded with several basic personalities upon installation, which are listed below. Users also have the flexibility to add, edit, and delete personalities, allowing for a personalized experience tailored to their specific needs. The initial personalities provided are:

- Factual Assistant
- Creative Guide
- Technical Expert

In addition to their individual system messages, each personality is configured with a preset temperature setting, labeled as "creativity" for the user, to further influence the LLM's language and response style.

Override Default Creativity

Each chosen personality comes with a default temperature setting, which is typically low for personalities like the Factual Assistant but considerably higher for the Creative Guide. However, users have the flexibility to temporarily adjust the level of creativity during a conversation, even while maintaining the selected personality. This is achieved by opting out of the default temperature and using a slider to set their preferred value between 0 and 1.

Reference Sources

To build user trust and ensure the chatbot's accountability for the information it provides, a list of references is included by default at the end of each generated answer. This reference list is presented as an enumerated list, detailing the name and location of each source document. The order of these references is determined by the earlier relevance ranking performed by the LLM component of the chatbot, irrespective of whether the actual response directly utilized information from every listed source. Should the user fully trust the chatbot and prefer not to see these references, this setting can be easily toggled off.

3.4 Components

The components work individually but has to be integrated in a smooth and simple way to ensure minimal amount of data loss in the pipeline. The components used in the final RAG framework of this project can be found in this section.

3.4.1 Knowledge base

The knowledge base of the chatbot consists of three parts, a searchable vector space, an information database and a map between index in the vector database and the information database.

Vector space

The vector space is implemented using the FAISS Python library, developed by Meta AI. FAISS is designed for efficient similarity search and clustering of dense vectors, particularly at scale [39].
A key strength of FAISS is its ability to perform high-speed nearest neighbor searches, even when the entire vector index does not fit within RAM. To achieve this, FAISS

relies on approximate nearest neighbor (ANN) algorithms instead of exact k-nearest neighbor (KNN) searches. According to the documentation, this design enables up to ten times faster queries and ten times lower memory usage, at the cost of approximately 10% reduction in accuracy [39].
Similarity search in FAISS is by default based on Euclidean distance and returns the index and distance of the top k vectors most similar to a given vector. This capability is particularly useful in RAG systems, where fast and scalable embedding-based retrieval is a core requirement.

Information database

The information database is implemented as a nested dictionary structure, serialized and stored between sessions as a JSON file. This structure is visualised in Figure 3.3. Each entry corresponds to a chunk of a document, identified by a unique key composed of the document name and chunk number (e.g., "document x – Chunk 10").
The value associated with each key is another dictionary containing metadata fields associated to the information as well as the information itself. The structure enables efficient lookup of information as well as its associated metadata for traceability.

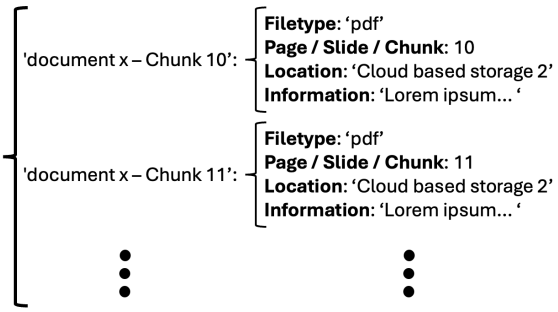


Figure 3.3: The figure shows the structure of the information database, a nested dictionary connecting the key of the information to the information and its metadata.

Mapping FAISS Index to Original Information

To effectively bridge the functional gap between the numerical vector representations stored in the FAISS index and the corresponding original textual information, an additional dictionary-based mapping system was implemented. FAISS efficiently handles the storage and retrieval of high-dimensional vectors, yielding indices of the most similar vectors. However, these indices alone do not directly link back to the full content of the documents from which the vectors were derived.
This mapping involved creating and maintaining a separate dictionary where each FAISS index was directly associated with its respective unique identifier pointing to the original information. This approach offered superior robustness, particularly when the knowledge base or the FAISS index underwent updates. By introducing this intermediate mapping layer, the system avoided a direct, rigid link between

the FAISS vector index and the raw information's storage location. This abstraction ensured that modifications to either the vector database (e.g., re-indexing, addition/deletion of vectors) or the information repository (e.g., updating source documents) could be managed independently with minimal disruption to the overall retrieval process. This design choice contributed to the system's flexibility and maintainability, ensuring accurate retrieval of content corresponding to the similarity search results.

Distribution of Data

In the knowledge base, approximately 96% of the data originates from images, with the remaining 4% derived from text. The image data is sourced from both PowerPoint slides and images extracted from PDF files, whereas text data is exclusively parsed from PDF documents

3.4.2 Retriever

The retriever builds on previously introduced components and performs multi-stage filtering to identify the most relevant information for a given query.
It begins by using the same encoder employed for document embedding, all-MiniLM-L6-v2 which produces dense vector representations of dimension 384. When a query is received, it is similarly encoded into a 384-dimensional vector. FAISS is then used to perform an approximate similarity search, returning the 20 nearest neighbors based on vector similarity.
To improve precision, the results are further refined using a cross-encoder. This model re-evaluates the top 20 candidates by scoring them in pairs with the original query, ranking them by contextual relevance. The five highest-scoring results are selected for further processing.
These five candidates, still represented as embedding indices, are then mapped back to their corresponding text entries in the information database via an index-to-information lookup. Each retrieved text chunk is subsequently scored for relevance by the same LLM used in the generator. The model assigns a score from 1 to 5, where 5 indicates high relevance and 1 indicates irrelevance. Any document scoring 2 or below is discarded from the final context used for response generation.

3.5 Challenges and Technical Decisions

To ensure a finished product within the limited time frame and the companies guidelines, decisions had to be made that highly impacted the outcome. These challenges and decisions are found in this secition.

3.5.1 Non-standardized PowerPoint

The PowerPoints used in the database are created by different individuals and therefore parsing text and images differently made it hard to collect structured information that can catch the concepts and essence of the slide. This due to context of data

being lost when the location is of high importance, for example an arrow pointing at something being an input by itself. This issue ended with the choice to convert all slides to images and feed them through an image pipeline which resulted in fewer entries while also increasing the relevance of each data point.

3.5.2 Document Extraction from Visual Layouts

Another significant challenge involved extracting structured information from a collaborative visual planning tool, similar to a digital whiteboard or mind map, that was used in the organization's workflow. Unlike traditional documents, the content was exported in a purely visual format, making it incompatible with standard text-based chunking and indexing methods commonly used in RAG systems.
To address this, a custom pre-processing pipeline was developed. The core idea was to identify semantically related regions based on visual proximity. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) was applied to group elements that were spatially close, based on the assumption that proximity implied contextual relevance. Once the clusters were identified, they were cropped from the original PDF with an added margin to preserve surrounding context and then converted to high-resolution PNG images for further processing.
This method proved effective and enabled the successful integration of content from the visual planning tool into the chatbot's knowledge base. The solution illustrates how visual artifacts from collaborative tools can be adapted for use in text-based retrieval systems through spatial clustering and image-based preprocessing strategies.

3.5.3 Access limitations

The original design of the chatbot envisioned a centralized solution accessible to users within the Volvo GTT's ecosystem via a web-based interface. The chatbot was intended to automatically synchronize and update its knowledge base in response to changes in the cloud-based document storage system used within the organization.
However, during development, two major limitations became evident. First, access to the necessary cloud storage platform could not be obtained within the scope of this project. Second, it became clear that deploying a centralized, internal web application for users within the Volvo Group ecosystem was also not feasible within project constraints.
These limitations had a significant impact on the system architecture. Due to the lack of cloud storage access, the chatbot could no longer dynamically update its knowledge base based on cloud data; instead, the knowledge base had to be updated manually. The absence of a centralized hosting solution also meant that the chatbot had to be run locally on each user's machine.
To mitigate these challenges, the chatbot was implemented as a standalone local application, including all necessary components except for the LLM, which continued to be accessed via an external API. This design ensured full core functionality, albeit with reduced scalability and ease of maintenance. While each user has the capability to update their own knowledge base, the recommended method is for one user to

update it and distribute the updated version to others.

4
Methods

This thesis employed an applied research approach to develop a functional RAG chatbot for the department at Volvo GTT. Applied research was chosen because its primary aim is to solve practical, real-world problems by utilizing existing theories, tools, and technologies. Given the extensive existing research, readily available tools, and robust frameworks for RAG chatbot development, this approach was the most suitable for achieving the project's goal of creating a working application. Consequently, the success of this project will be primarily evaluated on the system's functionality and user experience, rather than on academic innovation.

4.1 Development process

The development of the RAG chatbot followed an iterative process, grounded in the principles of agile software development. The project commenced with a concise literature review to gain insights into relevant research, existing frameworks, and best practices in RAG systems. This initial phase was followed by discussions with the AI Core Team at Volvo Group to assess available internal tools and explore potential external technologies. The findings of this exploratory stage significantly influenced the architectural design of the RAG system.
Following the design phase, each component of the RAG architecture was constructed and evaluated in isolation. This process leveraged existing and readily available tools and technologies, ensuring adherence to the project's constraints and goals.
Throughout the development lifecycle, continuous dialogue was maintained with supervisors and stakeholders at Volvo GTT through weekly meetings. Close collaboration with an AI Team at Volvo GTT also provided ongoing support. These interactions facilitated regular feedback, which in turn shaped the project's expectations and allowed for dynamic adjustments to the system's design and functionality in response to evolving user needs. Development priorities were consistently determined based on feasibility, user relevance, and the feedback received. This workflow fostered a dynamic and user-centered development process, emphasizing functionality and user experience, which directly aligns with the project's evaluation metrics.

4.2 Data collection

Data collection was conducted in accordance with the guidelines and limitations set by the supervisors at Volvo GTT. All files from two distinct areas of research

within the department were gathered and processed, as outlined in Section 3.1. No data from the designated areas were intentionally excluded, however, files that were ineligible for parsing were automatically filtered out during processing. Data security was maintained in compliance with Volvo GTT's policies, and no files were accessed or transferred outside of Volvo's secure networks.

4.3 Evaluation Strategy

The evaluation of this project's RAG system was structured around four key areas, identified as critical for assessing the performance of such an application: Information Retrieval, Generation, Efficiency, and Usability.

Each of these areas directly corresponds to aspects that received significant focus throughout the project's development. It is important to note that the underlying LLMs that power the RAG system were not subjected to a separate, in-depth evaluation. This decision was made because no direct development or modification of these foundational models was possible within the scope of this project, rendering their individual assessment largely irrelevant to the project's specific objectives.

4.3.1 Information Retrieval

Information retrieval is fundamental to all RAG systems. This evaluation aimed to quantify the effectiveness of the retriever component in identifying relevant information and, in this project's context, filtering the data used to formulate user responses and provide references. Given its critical role, three distinct evaluation methods were designed to assess the retriever's performance. For this purpose, a set of ten diverse queries was formulated, each with five variations, resulting in a total of 50 queries used for the retriever's comprehensive evaluation which can be found in Appendix A.2.

Prompt Sensitivity

The first and most straightforward evaluation of the retriever assessed its robustness to variations in query phrasing. The core principle behind this test was to determine how significantly the phrasing of a question, while maintaining the same intent, would impact the selection of source documents from the knowledge base. This is crucial because the retriever relies on similarity matching to locate relevant information. To measure this, each of the ten core questions was submitted to the retriever, along with its five variations. The sources linked to each query variation were tracked. A robust retriever would consistently identify the same set of sources across all variations of a single question, whereas a poor performance would be indicated by a significantly different selection of sources for each variation of the query.

Number of Sources

Optimizing the number of retrieved sources is crucial for RAG system performance. While providing more context generally leads to better LLM-generated answers, there are practical limitations. The LLM's context window imposes a constraint on the amount of information it can process. Additionally, an excessive number of sources can introduce redundancy, where multiple documents convey the same information, or, conversely, present overly diverse and potentially irrelevant content. The challenge lies in striking a balance: providing sufficient context to enable a comprehensive answer without overwhelming the LLM and diverting its focus from the primary query.
To address this, the ten core queries were again utilized with the retriever. This time, the evaluation focused on varying the number of sources that the different components of the retriever were instructed to identify, retain, or filter out. The quality of the identified sources was then assessed by assigning each source a score from 1 to 5, based on its relevance, redundancy, and similarity to the query's intent. While the time aspect of varying the number of sources was also considered, its detailed analysis is presented in a separate evaluation section.

Source Filtering

The process of identifying relevant sources within a RAG system involves applying filters of varying granularity to refine the selection of information. This refinement aims to provide the LLM with the most optimal context for accurately answering user queries.
This evaluation served as a direct continuation of the previous assessments concerning information retrieval, utilizing the same set of queries. The three distinct filters employed within the retriever were engaged to identify sources as they normally would. However, unlike standard operation, the sources that would typically be discarded by these filters were retained for analysis. A subsequent comparison was then conducted to assess the agreement or disagreement among how each filter judged individual sources. This analysis was crucial for identifying potential areas where the filtering structure could be optimized to improve overall retrieval performance.

Consistency

Consistency of all components needed to be ensured. To test this, the same query was given to the chatbot 500 times, then the result was analyzed to see if cross-encoder score, relevance score and consistency in retrieval deviated from run to run. This method ensured that the same query consistently end up with the same sources and therefore a similar answer.

4.3.2 Generation

The generation component of the RAG system is inherently dependent on the preceding information retrieval step. This evaluation focused on how effectively the language model interprets the retrieved context and generates responses that are

both relevant to the user's query and faithful to the underlying source material. To evaluate this, 30 questions were given to the RAG system and the answers were judged manually to assess if the answer was grounded or at least supported by the retrieved information or made up. The questions were of two natures, half were Q/A focused were the questions asked about very specific information, and half were exploratory where the chatbot should make assumptions and reason based on the retrieved information. The queries used in this evaluation can be found in Appendix A.1.

4.3.3 Efficiency

A recurring concern identified during the literature review for this project was the latency of RAG systems. The user's interaction with a RAG system involves two sequential steps: the identification of relevant information, followed by the generation of an answer by the language model. While the answer generation time is largely dependent on the underlying LLM and thus difficult to optimize without modifying the LLM itself, the retriever's performance is influenced by two key factors: the size of the knowledge base and the efficiency of its search mechanisms, alongside the process of distilling identified sources before providing them to the LLM. To facilitate a fluid conversational flow between the user and the RAG system, both generation and retrieval must be sufficiently fast. However, given the project's constraints, the optimization efforts and subsequent evaluation were exclusively focused on the retriever's efficiency.

4.3.4 Usability

The functionality and user-friendliness of the application were primarily assessed through direct user feedback collected during weekly supervisor meetings; however, this qualitative data is not easily quantifiable. To complement this feedback and provide a measurable outcome for this crucial aspect of the application, a questionnaire was administered to the initial users.
This questionnaire found in Appendix A.3 allowed users to rate specific aspects of the application on a Likert scale of 1 to 5, where 1 indicated the lowest satisfaction and 5 signified complete satisfaction. Responders were also encouraged to provide qualitative comments for each graded aspect, offering a more comprehensive understanding of their user experience.

4.4 Ethical Considerations

The development of a RAG system within a company operating in the EU necessitates careful consideration of its ethical implications and potential societal impact. This assessment has focused on three critical areas: transparency and explainability, security, and privacy concerns related to personal information. While transparency and security are common ethical considerations for AI systems, the importance of adhering to General Data Protection Regulation (GDPR) standards regarding personal data privacy is of outmost importance in this context.

4.4.1 Compliance with GDPR

To ensure compliance with the EU GDPR as well as the company's internal policies on data protection, several key measures were taken throughout the project.
The first measure was the careful selection of data sources. All documents used in the RAG system's knowledge base were retrieved from secure, closed, cloud-based platforms internal to the company. These platforms are assumed to meet both company security standards and GDPR requirements. Additionally, the dataset was intentionally curated to exclude potentially sensitive materials, such as internal interviews or reports containing personal data. Given the limited volume of such documents, exclusion was a straightforward and effective strategy for risk mitigation.
When metadata was extracted from documents, personally identifiable information, such as author names and contact details, was explicitly excluded. A residual risk remained for cases where document authors had embedded their names or contact information within the body of the documents themselves. Although the likelihood and severity of leakage in these cases was assessed to be low, an extra safeguard was implemented: the language model was instructed via system prompts not to disclose names, email addresses, or other personal identifiers if present in the retrieved context.
To further limit the risk of privacy breaches, users are informed through documentation that attempts to request personal data from the chatbot are not permitted.
Finally, the application operates locally on the user's machine. Aside from LLM queries made via an external API, no data is transmitted or synchronized externally. Chat histories and user-defined settings are stored locally, ensuring that users retain control over the data they choose to share with the system.
Together, these measures establish a practical and responsible data handling approach, consistent with both GDPR and internal corporate policies.

4.4.2 Data Security

Data security and the prevention of information leakage were central concerns expressed by the company throughout the development of the application. Accordingly, several precautionary measures were implemented to mitigate these risks.
The most effective control measure is the restricted availability of the application. It is designed to function only within the Volvo GTT's internal network and relies on LLMs hosted securely on this network. This ensures that the application cannot be used outside of the company environment, significantly reducing the risk of unauthorized access or data exfiltration.
Secondly, careful consideration was given to the sensitivity of the data made accessible to the RAG system. To ensure alignment with organizational data protection standards, the project restricted its scope to information deemed appropriate for use in AI applications. Rather than initiating a separate security review process, the data sources were selected based on existing guidelines and approvals for internal AI usage, thereby maintaining compliance while streamlining development.
Third, all intended users of the application already possess access to the included documents via existing channels, such as the company's internal cloud-based storage system. Since the RAG application merely facilitates a new way of interacting

with the same information, rather than exposing previously restricted data, risks commonly associated with prompt injection, jailbreaking, over-retrieval, or context window manipulation are rendered largely irrelevant in this context.
Finally, an additional safeguard was implemented to mitigate context window exploitation. Retrieved documents are filtered based on relevance, and only those with sufficient contextual alignment to the user's query are passed to the LLM. Furthermore, any retrieved context is discarded immediately after the response is generated. This ensures that sensitive information cannot be accessed or referenced in subsequent user prompts unless it is re-retrieved due to query relevance.
Collectively, these measures align with the company's security requirements and ensure that the RAG system operates within a secure, compliant, and controlled environment.

4.4.3 Transparency and Explainability

Although certain components of the system—most notably the underlying LLM, operate as black boxes and cannot be fully interpreted by users or developers, specific measures were implemented to enhance the overall transparency and explainability of the application, particularly in relation to its retrieval mechanism.
First, the application includes a user-configurable option to display source references. When enabled, each chatbot response is accompanied by a list of the retrieved documents that contributed to the answer. These references include the document title, the page number when applicable, and the storage location from which the document was retrieved. This feature allows users to independently verify the information presented and trace it back to the original context.
Second, efforts were made to communicate clearly how the retrieval system functions. Both in internal presentations and within this report, particular emphasis was placed on explaining the mechanism by which user queries are converted into vector embeddings and compared against the embedded document database. This contrasts with more intuitive, keyword-based search systems; for example, retrieval is based on semantic similarity rather than document titles, authors, or metadata. While this process may not be immediately apparent to end-users, transparency in its communication fosters a better understanding of the system's behavior and limitations.
Third, users are granted full access to the system's persona configurations. These system profiles allow users to select, modify, or create new personalities that define how the chatbot behaves and communicates during interactions. By making this component customizable, the application offers an additional layer of transparency and control over the conversational dynamics and tone of responses.
Together, these features aim to improve user trust, promote responsible usage, and support informed decision-making by providing insight into how the system operates and where its responses originate.

5

Results

This chapter presents the results of the evaluation of the RAG chatbot system. The results are organized into four main sections: information retrieval, generation, efficiency, and usability. The information retrieval section analyzes the impact of prompt sensitivity, the number of retrieved sources, and source filtering on retrieval quality. The generation section evaluates the quality of the model's output based on the retrieved content. The efficiency section focuses on the performance of system components and the impact of query variability. Finally, the usability section highlights practical insights for end users.

5.1 Information retrieval

As the retriever is the most fundamental part of a RAG chatbot, the relevance of the sources retrieved needed to be tested. The results below display how the retriever behaves to different types of queries and also how the scores are used to provide the correct sources.

5.1.1 Sensitivity of User Query

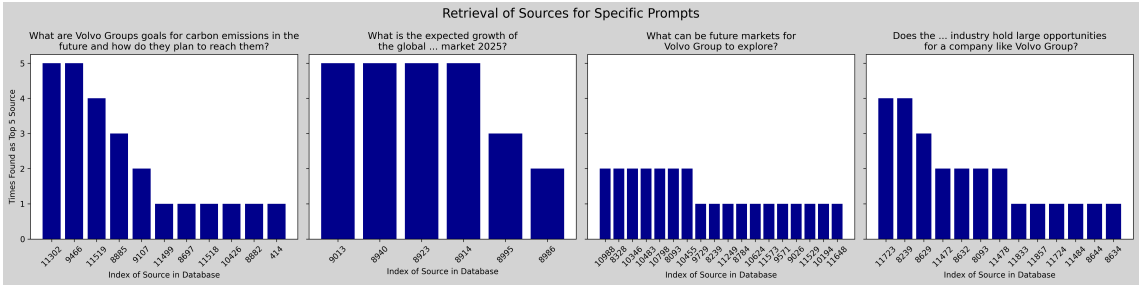


Figure 5.1: Distribution of retrieved sources for four distinct user queries. Each bar represents a unique document from the knowledge base, and its height indicates the frequency of its retrieval across five paraphrased versions of the same query.

Figure 5.1 illustrates the varying distribution of retrieved sources across several domain-specific user queries, highlighting the critical role of query specificity in effective information retrieval. The results demonstrate that highly specific queries directly correlate with a more concentrated retrieval of relevant information, indicating the importance of precise user input for accurate RAG performance. It also

shows that when the query was paraphrased, some domains retrieve new sources while some still retrieve the same ones.

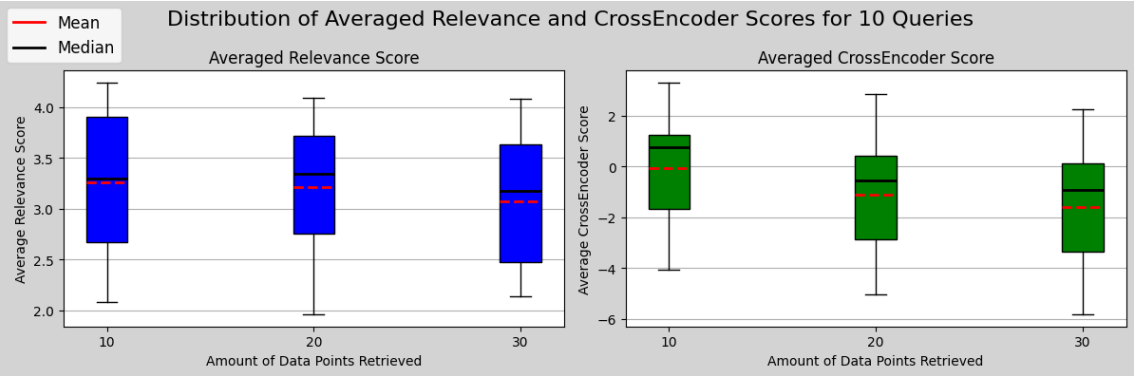


Figure 5.2: Plot of Averaged Relevance Score (to the left) and Cross-Encoder Score (to the right) for 10 queries paraphrased into 50 in total. The x-axis shows the amount of initially retrieved sources from the knowledge base.

By looking at the 10 different query categories described in Section 4.3.1, Figure 5.2 shows how the 50 queries are distributed when it comes to relevance score and score from the cross-encoder. The figure indicates that the relevance doesn't correlate with how many sources that are provided. However, the cross-encoder score is lowered quite significantly on average when more sources are retrieved from the knowledge base, indicating fewer sources to be better.

5.1.2 Consistency

Table 5.1: Table displaying the consistency of the retriever. Average is collected for 500 runs and compared to first run. Also, it is displayed how many times the same source was found.

Index	1st Rel	Avg. Rel	1st Cross	Avg. Cross	Runs Found
11302	2.0	2.004	6.4363	6.4363	500
11499	5.0	5.0	4.4612	4.4612	500
11519	5.0	5.0	6.8492	6.8492	500
6729	4.0	4.0	5.0713	5.0713	500
9466	2.0	2.0	6.8859	6.8859	500

Table 5.1 shows the consistency for the different retriever components. By collecting values from the components for 500 runs with the exact same query, it clearly displays the deterministic behavior of the components by outputting the same values over 500 runs. Only deviation is for the average relevance score where, when checked manually, it shows that for two runs, the LLM gave source 11302 a relevance score of 3.0.
The deterministic nature of the bi-encoder and cross-encoder is confirmed. The minor stochasticity observed in the LLM-based relevance judgment (2 out of 500 runs

for index 11302) is likely due to non-deterministic behavior in the LLM's inference process, even at low temperatures, and does not significantly impact overall system reliability.

5.1.3 Number of Sources

Referring back to Figure 5.1, the number of sources retrieved based on the query illustrates that no specific optimal number of sources exist for all queries, but is rather dynamic based on query.

5.1.4 Source Filtering

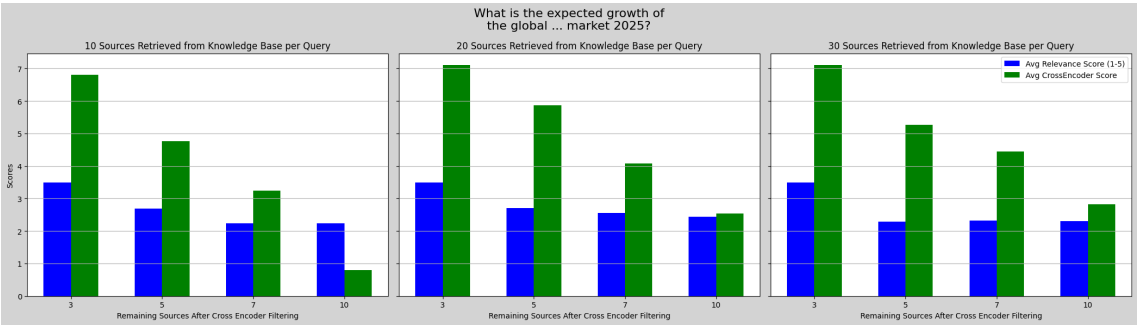


Figure 5.3: Referring back to Figure 5.1, the plot shows cross-encoder and relevance score for different number of sources retrieved (per plot) and cross-encoded (x-axis). This plot is based on a query with high density of sources retrieved

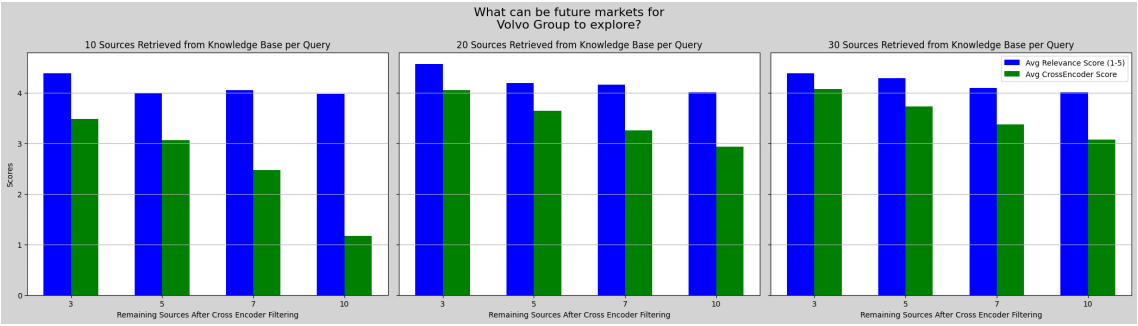


Figure 5.4: Referring back to Figure 5.1, the plot shows cross-encoder and relevance score for different number of sources retrieved (per plot) and cross-encoded (x-axis). This plot is based on a query with low density of sources retrieved

A comparison of Figure 5.3 and Figure 5.4 reveals distinct patterns in source retrieval and cross-encoder scoring influenced by the initial user query. For queries like "What is the expected growth of the global ... market 2025?" (Figure 5.3), which Figure 5.1 shows have a high concentration of very few dominant relevant sources, the average cross-encoder score increases or stays the same as the number of initial sources retrieved from the knowledge base increases from 10 to 30. However, the

cross-encoder score of the sources remaining after the filter significantly drops as the amount increases from 3 to 10.
Conversely, for queries such as "What can be future markets for Volvo Group to explore?" (Figure 5.4), corresponding to a more diffuse distribution of relevant sources in Figure 5.1, the cross-encoder score significantly increases from 10-20 sources and then flattens out for 30. When the cross-encoder allow more sources after filtering, the cross-encoder score drops but the same behavior is experienced for 20 and 30 sources.
Across both queries and varying initial source counts (10, 20, or 30 sources), the average relevance score shows only a minor reduction as the number of sources remaining after cross-encoder filtering increases.

5.2 Generation

Table 5.2: Categorization of queries into Q/A and Explore with Yes/No counts

Category	Yes	No
Q/A	13	2
Explore	13	2

Table 5.2 displays how different categories of queries makes the LLM base its answers on the sources provided from the retriever. The queries asked can be found in Appendix A.2 and it is clear that the nature of the questions is of higher significance than the "category". Due to confidentiality, no full queries nor any answers are displayed in the report. It should also be stated that the test only analyzed if the LLM used the sources provided to answer the query and not if it answered them well. In some cases the LLM used the sources to answer that the information was not relevant which shows that an irrelevant source leaked through the retrieval system. However, some questions where answered very well and with a lot of good ideas without a clear ground in provided sources.

5.3 Efficiency

The results below display the efficiency of the retrieval architecture and was used to decide the amount of data points that would be retrieved. This analysis was performed to find the most amount of information retrieved in as short amount of time as possible.

5.3.1 Component analysis

When measuring time for retrieval for one single query, Figure 5.5 and 5.6 shows the distribution over 500 runs. Figure 5.5 shows that even though the same query has been used, the time is distributed between approximately 1.9 seconds and 3.6 seconds with outliers on the higher end.

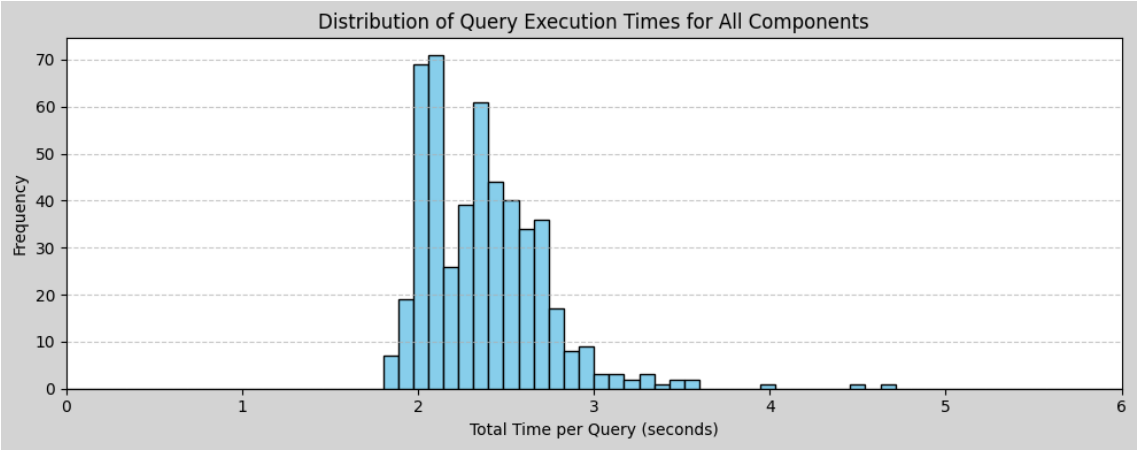


Figure 5.5: Distribution of Execution times for retrieval measured as a total for all retrieval components and 500 runs. Same query was used for all runs.

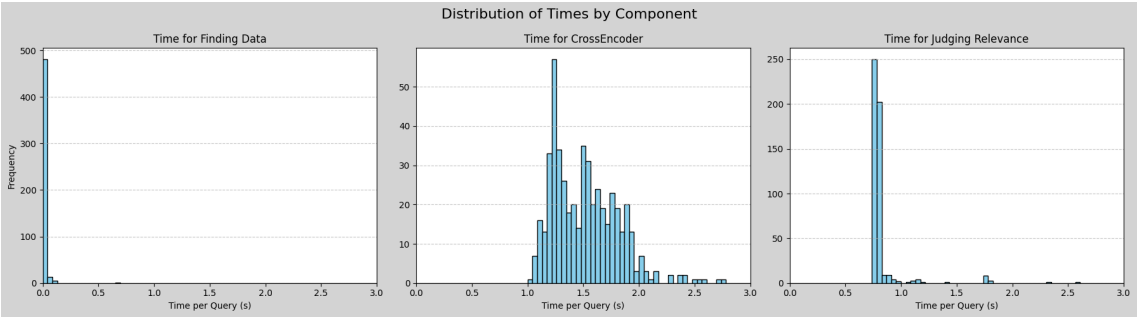


Figure 5.6: Distribution times for retrieval measured for all individual components in the retriever and 500 runs. Same query was used for all runs.

For deeper analysis, results from each component was taken, shown in Figure 5.6 which clearly states that the cross-encoder is the component showing a more stochastic behavior when it comes to execution time while the database search and relevance check behaves more linear in terms of time.
When looking at retrieval time for different queries, same set as in Section 4.3.1, it clearly displays that there is a linear behavior between retrieving more sources and total time. The cross encoder is only affected by the amount of sources retrieved from the knowledge base and the LLM judging relevance is only affected by the remaining sources after cross encoder filtering. This is expected behavior.
Figure 5.7 displays the time for retrieval when different amounts of data was retrieved. The result from this was used to decide how many data points that should be provided to the LLM for answering.

5.4 Usability

User satisfaction with the chatbot's retrieval capabilities and its personality features was generally positive, though with identifiable areas for potential refinement. The effectiveness of information retrieval and its subsequent utilization in answer for-

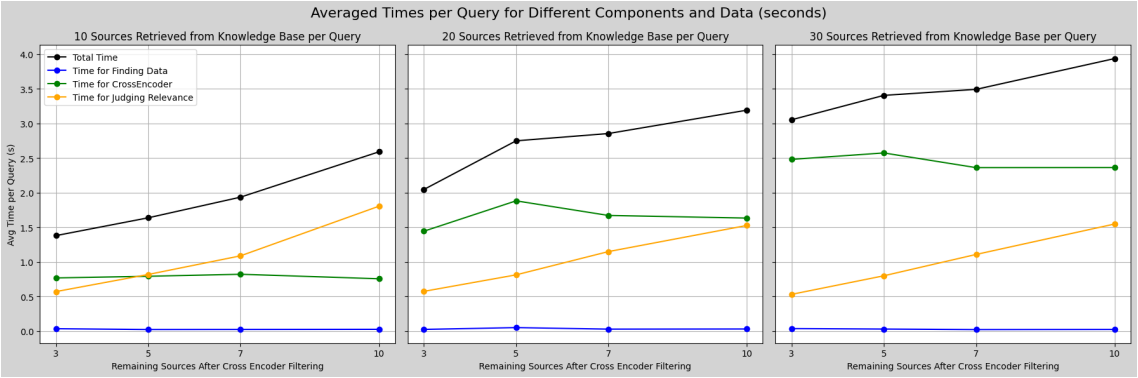


Figure 5.7: Time as a function of remaining sources after filtering for different numbers of sources retrieved from the knowledge base. The lines represent the different components of the retriever with black representing a total time.

mulation received satisfactory remarks, with users appreciating the system's ability to find relevant documents and integrate this information into responses. However, feedback indicated that the accuracy and relevance of answers were highly dependent on the specificity of the user's query, with broader questions sometimes yielding less precise results. This suggests that while the RAG mechanism generally performs well, its optimal performance is influenced by both user input quality and the comprehensiveness of the underlying data sources.
Regarding the chatbot's "personalities," users expressed satisfaction with the observed behavioral changes and adherence to instructions, indicating that the personality configurations effectively altered the chatbot's behavior. Users found that the system clearly followed the specified personality instructions, leading to a noticeable difference in interaction style.

6
Discussion

This chapter provides an in-depth analysis and interpretation of the empirical findings derived from the evaluation of the RAG chatbot system, as detailed in Chapter 5. The discussion contextualizes these results within the broader objectives of this thesis, to democratize corporate information through multimodal RAG, and relates them to existing literature. We critically examine the performance of the system's core components, explore the implications of our design choices, and reflect on the system's overall efficacy and practical utility.

6.1 Interpretation of Information Retrieval Findings

The efficacy of the RAG system is fundamentally tied to its ability to accurately and efficiently retrieve relevant information. Our findings regarding information retrieval illuminate several key aspects of the system's performance, particularly concerning the precision and contextuality of the retrieved chunks.

6.1.1 The Critical Role of User Query Specificity

As demonstrated in Chapter 5, the specificity of the user query proved to be the most significant factor influencing the relevance and composition of retrieved information. This outcome aligns with expectations given the deterministic nature of the dense retrieval process, where the LLM's generative output is directly dependent on the quality and specificity of the information provided. The knowledge base, exclusively populated with domain-specific data from the two departments, led to variations in answer relevance directly correlated with the density in the vector space of available information for a given topic. Specifically, within densely populated regions of the FAISS index corresponding to extensively researched topics, even minor variations in query phrasing could lead to shifts in the search vector, consequently yielding a different set of semantically similar documents via ANN search.
The paraphrasing of identical queries was used to simulate diverse user formulations, with the aim of ensuring consistent source retrieval regardless of phrasing. However, this objective was not consistently achieved across all test cases, particularly revealing sensitivities in the retrieval mechanisms when confronted with vague or abstract queries concerning exploratory themes like opportunities and exploration. While semantic matching aims to capture conceptual similarity despite variation in keywords, its strength can become a challenge with to vague queries. In such instances,

the broad semantic space associated with a vague term may yield multiple relevant, yet distinct, sets of documents, making it difficult for the system to pinpoint a singular 'best' set without further contextual cues. However, as evidenced in Section 5.2, the LLM's generative output remained of high quality. This indicates that while the specific retrieved sources might vary due to query nuances, the semantic matching successfully identified functionally equivalent or sufficiently relevant alternative contexts, thus maintaining the overall quality of the generated answers. This underscores the robustness of semantic matching as a retrieval strategy, capable of providing effective contextual grounding even with user-varied query formulations.

6.1.2 Consistency and Determinism of the Retriever

The combination of FAISS for initial retrieval and the all-MiniLM-L6-v2 cross-encoder for re-ranking demonstrated a high degree of consistency and determinism in identifying the top 5 most relevant chunks. For identical queries, the system consistently returned the same set of re-ranked documents, affirming the reliability of this two-stage retrieval process. The LLM-based relevance judgment, while introducing a qualitative assessment, was observed to further refine this consistency by actively filtering out context that, despite high semantic similarity, might be deemed less relevant by the LLM's advanced contextual understanding. This iterative filtering process aimed to balance quantitative retrieval metrics with qualitative relevance for the final generation.

6.1.3 Optimizing the Number of Sources and Filtering Strategies

The adopted strategy of retrieving 20 initial documents for later re-ranking to the top 5 was based on the premise that the all-MiniLM-L6-v2 cross-encoder would provide a more nuanced and accurate relevance assessment compared to the initial encoder's semantic matching. This approach aimed to leverage the broader recall of the initial ANN search, ensuring that a diverse set of semantically relevant, though not necessarily perfectly matching, documents were identified. The cross-encoder would then precisely discriminate the most relevant among these for the final top 5. While the results largely supported this decision, they also indicated a marginal increase in re-ranking accuracy with a greater number of initial sources provided to the cross-encoder. However, the performance loss observed when reducing the initial retrieval count from 30 to 20 sources was not substantial, rendering computational efficiency a critical deciding factor for the chosen retrieval volume. This trade-off is further elaborated in Section 6.3. Furthermore, the LLM-based relevance judgment, applied to the top 5 sources, did not yield a statistically significant change in overall answer quality when tested with varying total amounts of input data beyond this subset.

6.2 Analysis of Generation Quality

The ultimate measure of a RAG system's effectiveness lies in its ability to generate high-quality, relevant, and accurate responses. This analysis focused on how well the language model utilized the retrieved context to respond to two distinct categories of prompts: factual Q/A questions and more open-ended, explorative inquiries.
While the behavior and performance of a RAG system can be heavily influenced by the system prompt, the same prompt was used for both question types to ensure consistency during evaluation.
Although the assessment involved reading the LLM's responses in parallel with the retrieved sources, it remained somewhat subjective and, at times, unclear to what extent the sources influenced the generated answers. Nevertheless, it was evident that the majority of responses were well grounded in the retrieved context. On the rare occasions when the LLM strayed significantly from the sources, it was typically in cases where the question extended beyond the available knowledge base, and the retrieved context offered little to no value, prompting the LLM to speculate.
In a few instances, the model omitted critical details that were present in the sources, particularly in factual Q/A scenarios. However, even in these cases, the answers were still reasonable and relevant. Notably, the model never contradicted the retrieved sources, which is a strong indicator of effective mitigation of hallucination, a common challenge in generative systems.
In summary, the evaluation demonstrated that the RAG system performs well across different types of questions, effectively leveraging retrieved context in most cases. While there remains room for improvement, particularly in ensuring completeness and reducing occasional overgeneralization, the results highlight the system's strong potential and reliability.

6.3 Discussion on System Efficiency

Beyond accuracy, the practical deployment of a RAG system necessitates an understanding of its computational efficiency. This section discusses the performance characteristics of our pipeline components.

6.3.1 Execution Time Variability and Component Contributions

Given the findings in Section 6.1.3, which indicated no statistically significant accuracy gains when increasing the initial retrieval count from 20 to 30 sources, computational efficiency became a paramount consideration in optimizing the RAG pipeline's performance. Analysis of the pipeline's individual components revealed that the cross-encoder stage exhibited the highest degree of variability in execution time. This stochastic behavior can be attributed to the dynamic nature of pairwise relevance scoring, where processing time is influenced by the varying lengths and complexities of both the query and the retrieved chunks.
Furthermore, the execution time for both the cross-encoder re-ranking and the sub-

sequent LLM-based relevance judgment demonstrated a linear increase proportional to the number of sources being processed. This linear scaling justified a strategic focus on minimizing the number of retrieved sources without compromising the overall retrieval accuracy, aligning with the principle of optimal resource allocation. Consequently, 20 initial sources were determined to strike an optimal balance between ensuring a sufficiently broad initial search recall from the knowledge base and maintaining high system efficiency, as detailed in Section 6.1.3. The subsequent reduction to five sources for the LLM-based relevance check was similarly driven by efficiency considerations. Given the comparatively higher computational cost of LLM inference per source, this limited subset proved optimal for minimizing the latency associated with the final judgment phase, ensuring that only the most highly vetted content is processed by the generator. In contrast, the FAISS ANN search, benefiting from its optimized indexing and potential GPU acceleration, exhibited negligible execution time relative to the other pipeline stages, irrespective of the initial number of retrieved sources. This confirmed FAISS as an efficient backbone for large-scale initial document retrieval.

6.4 Discussion on Usability

The usability evaluation of the RAG-based chatbot, provides valuable insights into how such systems are perceived and utilized by users within an industrial context. The evaluation focused on users with varying levels of technical expertise and limited prior experience with chatbot technology, which makes the feedback particularly relevant for assessing accessibility and practical utility.
Overall, the responses suggest that the application is both usable and useful, though naturally perceived as a prototype and proof-of-concept. Users were generally positive about the relevance of information retrieved by the system and how it was integrated into generated responses. However, the evaluation also revealed areas where improvements could enhance the system's reliability, transparency, and robustness.

6.4.1 User Perception and Practical Insights

Users reported high satisfaction with the RAG system's ability to retrieve and utilize relevant documents when answering queries. The system was frequently described as "polished" and "useful," even at this early stage. This suggests that the core retrieval functionality already delivers tangible value, particularly in domains where structured document access is otherwise time-consuming.
That said, users also highlighted the dependency of answer quality on the specificity of the question and the comprehensiveness of the underlying document base. Broader or ambiguous queries occasionally yielded less accurate or incomplete responses. This reflects a fundamental limitation of similarity-based retrieval methods in dealing with underspecified or abstract prompts and emphasizes the importance of well-structured, high-quality source content.
The inclusion of source references and the ability to inspect retrieved documents directly contributed positively to user trust and transparency. This feature was

especially important given that users expressed difficulty distinguishing between model-generated content and retrieved knowledge. While this opacity is an inherent limitation of LLM-based systems, the implemented reference system appears to mitigate some of that concern.
Personality customization also played a role in shaping the user experience. Respondents found that the personalities effectively influenced response tone and behavior, although some noted that comparing them meaningfully requires more extensive use. This suggests that while personalization adds flexibility, it may also introduce cognitive overhead for new users.
In summary, the feedback indicates that the RAG system is functional and well-received, though further development, especially around data curation, retrieval precision, and system feedback, will be essential for future deployment at scale. The project successfully demonstrates that RAG-based chatbots can offer usable and valuable support in knowledge-intensive settings, even for users with limited AI experience.

6.5 Limitations

Despite the promising results, this study acknowledges several limitations that constrain the performance of the developed RAG system. These limitations arise primarily from data scope, time constraints and the inherent complexity of evaluating AI systems.
First, the exclusion of certain data, such as confidential or GDPR-sensitive documents, as well as unsupported file types like audio or video, limited the comprehensiveness of the knowledge base. While the majority of relevant textual data was included, this exclusion may create blind spots, potentially leading users to assume the chatbot has access to more information than it actually does.
Second, while retrieval performance was generally strong, the system would have benefited from additional capabilities. These include alternative indexing methods, such as filtering by topic, author, or source, as well as the ability to narrow searches to specific subsets of the knowledge base. These features were not implemented due to time constraints but are important considerations for future iterations.
Third, the use of LLM-based relevance filtering proved to be an effective and novel strategy for refining the retrieval process. However, this approach introduces a layer of subjectivity and computational overhead. Its judgments may not always align with human expectations, particularly in cases where domain-specific or contextual nuances are required to assess relevance accurately.
Together, these limitations highlight areas for improvement and guide future development efforts aimed at enhancing the system's precision, flexibility, and reliability in real-world applications.

7
Conclusion

A RAG chatbot can be developed to improve access to unstructured information for end users in an industrial setting like Volvo GTT by designing a robust system that efficiently processes diverse corporate data and delivers grounded, verifiable responses. This is achieved through an approach involving a multimodal data extraction and processing pipeline, a multi-stage information retrieval mechanism, and a generative language model. Specifically, the development entails creating a system capable of extracting both text and images from complex, domain-specific documents like PDFs and PowerPoint presentations, transforming this unstructured data into a knowledge base where content is vectorized using embeddings and indexed efficiently with FAISS. The core of the access improvement lies in its multi-stage retrieval, which begins with broad initial recall via FAISS, followed by a refined reranking using a cross-encoder such as all-MiniLM-L6-v2, and ends in an LLM-based relevance judgment to ensure only the most relevant context is provided to the generator. This careful refining of information allows the LLM to synthesize high-quality, relevant, and well-grounded answers, effectively mitigating issues like hallucination and ensuring that responses are directly derived from the domain-specific corporate knowledge base.

The effectiveness of such a chatbot in improving information access for end users is thoroughly evaluated across several key dimensions. Evaluation demonstrates that user query specificity critically impacts retrieval precision, highlighting that while semantic matching generally handles varied phrasing, vague queries call for a more nuanced interpretation by the system. The multi-stage retrieval pipeline consistently proves effective in delivering accurate context, even with some variability in source selection for less specific queries, as the LLM reliably produces high-quality outputs. From an efficiency standpoint, while initial FAISS retrieval is rapid, the cross-encoder and LLM judgment stages are identified as primary contributors to latency, necessitating optimal tuning of source counts to balance broad search capabilities with computational efficiency. Crucially, user evaluation validates the system's utility and usability, with high satisfaction regarding information relevance and, most importantly, enhanced trust stemming from transparent source traceability. This mechanism directly addresses the challenge of verifying AI-generated information, making the system a reliable tool for democratizing internal corporate knowledge. Despite certain limitations such as data scope and fixed model choices, the development and rigorous evaluation demonstrate how a RAG chatbot can significantly enhance access to unstructured information, providing valuable insights for future advancements in industrial knowledge management.

Bibliography

[1] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[2] S. J. Mielke, Z. Alyafeai, E. Salesky, *et al.*, "Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp," *arXiv preprint arXiv:2112.10508*, 2021.

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

[4] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[5] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

[6] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[7] Milvus, *What are decoder-only models vs. encoder-decoder models?*

[8] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International conference on machine learning*, PMLR, 2017, pp. 1243–1252.

[9] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[10] J. Wei, X. Wang, D. Schuurmans, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[11] L. Ouyang, J. Wu, X. Jiang, *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[12] Z. Xu, S. Jain, and M. Kankanhalli, "Hallucination is inevitable: An innate limitation of large language models," *arXiv preprint arXiv:2401.11817*, 2024.

[13] J. Cheng, M. Marone, O. Weller, D. Lawrie, D. Khashabi, and B. Van Durme, "Dated data: Tracing knowledge cutoffs in large language models," *arXiv preprint arXiv:2403.12958*, 2024.

[14] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 13 693–13 696.

[15] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" In *Proceedings*

of the 2021 ACM conference on fairness, accountability, and transparency, 2021, pp. 610–623.

[16] N. F. Liu, K. Lin, J. Hewitt, *et al.*, "Lost in the middle: How language models use long contexts," *arXiv preprint arXiv:2307.03172*, 2023.

[17] P. Lewis, E. Perez, A. Piktus, *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[18] R. Sriramanan, A. Singh, T. Chhabra, A. Jain, and H. Mehta, "The Practical Applications of Retrieval-Augmented Generation in AI," *Journal of Computer Science and Software Development*, vol. 3, p. 205, Sep. 2024, Accessed: May 27, 2025. [Online]. Available: `https://www.jscholaronline.org/articles/JCSSD/The-Practical-Applications-of-Retrieval-Augmented.pdf` (visited on 05/27/2025).

[19] Microsoft. "5 key features and benefits of retrieval augmented generation (RAG)." Accessed: May 27, 2025. (Feb. 2025), [Online]. Available: `https://www.microsoft.com/en-us/microsoft-cloud/blog/2025/02/13/5-key-features-and-benefits-of-retrieval-augmented-generation-rag/` (visited on 05/27/2025).

[20] Trantor. "What is Retrieval Augmented Generation (RAG)?" Accessed: May 27, 2025. (2025), [Online]. Available: `https://www.trantorinc.com/blog/what-is-rag-retrieval-augmented-generation` (visited on 05/27/2025).

[21] K. Ma, A. Al-Hammami, and N. Vasanth. "RAG vs Fine Tuning: The Hidden Trade-offs No One Talks About." Accessed: May 27, 2025. (Feb. 2025), [Online]. Available: `https://b-eye.com/blog/rag-vs-fine-tuning/` (visited on 05/27/2025).

[22] Label Your Data. "RAG vs Fine Tuning: Which Method to Choose." Accessed: May 27, 2025. (Sep. 2024), [Online]. Available: `https://labelyourdata.com/articles/rag-vs-fine-tuning` (visited on 05/27/2025).

[23] Y. Gao, X. Ma, J. Lin, *et al.*, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.

[24] Y. Guo, M. Sun, J. Su, W. Xie, T. Luo, and H. Yuan, "Retrieval-Augmented Generation (RAG) for Large Language Models: A Survey," *arXiv preprint arXiv:2404.10982*, 2024.

[25] W. Fan, J. Huang, P. Zhou, X. Wang, and H. Zheng, "Retrieval-Augmented Generation (RAG) for LLMs: A Comprehensive Survey," *arXiv preprint arXiv:2310.02107*, 2023.

[26] Y. Esra, M. Benouis, A. Alhafdhi, and O. Boulares, "Towards Efficient and Robust Retrieval-Augmented Generation for Large Language Models," *arXiv preprint arXiv:2311.02641*, 2023.

[27] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 3981–3990, 2019. [Online]. Available: `[https://aclanthology.org/D19-1172/](https://aclanthology.org/D19-1172/)`.

[28] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 7, pp. 1741–1750, 2019.

[29] M. Wang, J. Ma, F. Yang, X. Lu, X. Li, and Y. Zhang, "Knowledge graph-augmented large language models: A survey," *arXiv preprint arXiv:2312.13882*, 2023.

[30] L. Zheng, W.-L. Chiang, Y. Sheng, *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 595–46 623, 2023.

[31] B. Ni, Z. Liu, L. Wang, *et al.*, "Towards trustworthy retrieval augmented generation for large language models: A survey," *arXiv preprint arXiv:2502.06872*, 2025.

[32] "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)." (2016), [Online]. Available: `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679` (visited on 05/29/2025).

[33] P. Voigt and A. Von dem Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing, 2017.

[34] I. Kotsyba and Y. Kotsyba, "AI and the GDPR: Navigating the Legal and Ethical Landscape," in *Artificial Intelligence Ethics and Governance*, Springer, Cham, 2023, pp. 165–188.

[35] C. Goh, H. Hada, A. Madaan, N. Rajani, and D. Yin, "Red Teaming Large Language Models: A Comprehensive Survey," *arXiv preprint arXiv:2311.08272*, 2023.

[36] H. Liu, X. Li, Z. Zhao, *et al.*, "Prompt injection attacks against ChatGPT," *arXiv preprint arXiv:2302.04018*, 2023.

[37] C. Gong, L. Hu, Y. Wang, Z. Zhang, G. Xu, and J. Wang, "Secure Large Language Models: A Survey," *arXiv preprint arXiv:2308.06674*, 2023.

[38] R. Dhamija, M. Jain, N. Kumar, and M. Sharma, "Comparison of Anonymization Techniques for Privacy Preserving Data Mining," in *2018 International Conference on Computing, Power and Communication Technologies (ICCPCT)*, IEEE, 2018, pp. 235–240.

[39] J. Johnson, M. Douze, and H. Jégou, *FAISS: Facebook ai similarity search*, `https://faiss.ai`, 2017.

A

Appendix 1

A.1 Queries for Generation test

The answers are graded from -1 to 1 where 1 means that the answer was based on the provided sources, 0 means that the sources were used but the answer also contained information from the LLM and -1 means that the answer was solely based on the LLM's knowledge.

Q/A Questions

Question of a question-answer nature where the answer is predicted to exist within the knowledge base.

1. What is the expected growth of the global [redacted] market 2025?
2. **Answer: 1**
3. What are Volvo Groups goals for [redacted] in the future and how do they plan to reach them?
4. **Answer: 1**
5. Who are Volvo Groups biggest competitors?
6. **Answer: 1**
7. What are the general prices of electricity in northern Europe?
8. **Answer: 1**
9. What are Volvo Group's largest connections to [redacted]?
10. **Answer: 1**
11. How many employees does Volvo Group have?
12. **Answer: -1**
13. How many vehicles are in Volvo Trucks fleet?
14. **Answer: 1**
15. How many different models does Volvo Trucks fleet include?
16. **Answer: 1**
17. Where does Volvo Group have offices in the world?
18. **Answer: -1**
19. Who is the largest [redacted] of Volvo Trucks?
20. **Answer: 1**
21. How many [redacted] does Volvo plan to launch?
22. **Answer: 1**
23. Are there any future plans to include [redacted] at Volvo Group?
24. **Answer: 1**
25. How does Volvo plan to address future [redacted] in their trucks?

26. **Answer: 1**
27. Which country has the highest amount of Volvo trucks?
28. **Answer: 1**
29. How large are the different parts within Volvo Group?
30. **Answer: 1**

Explorative questions

Questions where the answer might not be available in the sources and the response should be more explorative in nature, but the test evaluates if the answer is at all grounded in the sources provided to the LLM.

1. What can be future markets for Volvo Group to explore?
2. **Answer: 0**
3. Does the [redacted] hold large opportunities for a company like Volvo Group?
4. **Answer: 1**
5. Explain opportunities for Volvo within the [redacted] market
6. **Answer: 1**
7. Where geographically can the largest potential within the [redacted] sector be found for Volvo Group?
8. **Answer: 0**
9. What troubles can Volvo Group have when it comes to [redacted]?
10. **Answer: 1**
11. Can it be economically beneficial for Volvo Group to expand towards [redacted]?
12. **Answer: 1**
13. What opportunities can be found in [redacted] for Volvo Group?
14. **Answer: 1**
15. Are there any current business opportunities for Volvo Group in [redacted]?
16. **Answer: 1**
17. What future markets can Volvo Penta explore?
18. **Answer: 1**
19. What future businesses exists in [redacted] for Volvo Group?
20. **Answer: 1**
21. Predict Volvo Group's earning by 2050
22. **Answer: 1**
23. What can be future markets for Volvo Construction to explore?
24. **Answer: 1**
25. What pain points can you predict for Volvo group as a company in the next 5, 10 and 20 years?
26. **Answer: -1**
27. How would [redacted] affect Volvo group as a company?
28. **Answer: 1**
29. In an age where [redacted] is becoming more and more relevant, what potential is there for Volvo group to break in to this [redacted]?
30. **Answer: -1**

A.2 Queries Used for Retriever Tests

Used different combinations defined for each test to display results in Sections 5.1.1, 5.1.2, 5.1.3, **??**, 5.3.1 and **??**

Original Query 1:

- What is the expected growth of the global [redacted] logistics market 2025?

Paraphrased Queries:

- What is the projected size of the global [redacted] logistics market in 2025?
- How much is the [redacted] logistics industry expected to grow globally by 2025?
- What are the forecasts for the global [redacted] logistics market's expansion in 2025?
- Can you provide the anticipated growth figures for the worldwide [redacted] logistics sector in 2025?

Original Query 2:

- What can be future markets for Volvo Group to explore?

Paraphrased Queries:

- What new industries could Volvo Group expand into in the future?
- Which market segments might Volvo Group consider targeting next?
- Are there any emerging business areas Volvo Group is likely to explore?
- What future opportunities exist for Volvo Group outside its current lines of business?

Original Query 3:

- What are Volvo Groups goals for carbon emissions in the future and how do they plan to reach them?

Paraphrased Queries:

- What are Volvo Group's future objectives regarding carbon emissions reduction?
- Could you outline Volvo Group's plans for minimizing their carbon footprint in the coming years?
- How does Volvo Group intend to address carbon emissions in its long-term strategy?
- What sustainability targets has Volvo Group set to lower its greenhouse gas emissions moving forward?

Original Query 4:

- Does the [redacted] hold large opportunities for a company like Volvo Group?

Paraphrased Queries:

- Can the [redacted] offer significant growth potential for Volvo Group?
- What kind of opportunities could the [redacted] present to a company like Volvo Group?
- Is the substantial business potential for Volvo Group in [redacted] markets?
- How might Volvo Group benefit from developments in the [redacted]?

Original Query 5:
- Who are Volvo Groups biggest competitors?

Paraphrased Queries:
- Which companies are the primary rivals of Volvo Group in the heavy vehicle industry?
- Can you list the key competitors that challenge Volvo Group's market position?
- Who are the main industry players competing with Volvo Group?
- Identify the top companies that compete with Volvo Group in its sector.

Original Query 6:
- Explain opportunities for Volvo within the [redacted] market.

Paraphrased Queries:
- What potential does Volvo have to grow its business in [redacted] industry?
- How can Volvo Group tap into opportunities in the [redacted] supply chain?
- In what ways could Volvo benefit from involvement in [redacted] sector?
- What are the prospects for Volvo Group in supporting [redacted]?

Original Query 7:
- [redacted] can the largest potential within the [redacted] be found for Volvo Group?

Paraphrased Queries:
- In which [redacted] does the [redacted] offer the greatest potential for Volvo Group?
- [redacted] the [redacted] could Volvo Group find the most promising [redacted] opportunities?
- What [redacted] hold the highest growth prospects for Volvo Group in the [redacted]?
- Which [redacted] present the best opportunities for Volvo Group within the [redacted]?

Original Query 8:
- What troubles can Volvo Group have when it comes to [redacted]?

Paraphrased Queries:
- What challenges might Volvo Group face in [redacted]?
- Which obstacles could hinder Volvo Group's efforts to [redacted]?
- What are the main difficulties Volvo Group may encounter while [redacted]?
- How could the process of [redacted] pose problems for Volvo Group?

Original Query 9:
- Can it be [redacted] for Volvo Group to expand towards [redacted]?

Paraphrased Queries:
- Would [redacted] make [redacted] sense for Volvo Group?
- Is there [redacted] for Volvo Group in developing [redacted]?
- Could building out [redacted] be a [redacted] for Volvo Group?
- How might [redacted] benefit Volvo Group [redacted]?

Original Query 10:
- What opportunities can be found in [redacted] for Volvo Group?

Paraphrased Queries:
- How could Volvo Group tap into opportunities within the [redacted] sector?
- What potential business areas in [redacted] handling could Volvo Group explore?
- Are there growth prospects for Volvo Group in the [redacted] industry?
- What role could Volvo Group play in advancing solutions for [redacted]?

A.3 User feedback

A.3.1 Questionnaire for evaluation of application

Note: The application was named SAGE within the company and is referenced as such below.

Answer on a scale from 1-5 based on your satisfaction with SAGE where 1 is very unsatisfied and 5 is very satisfied. You will also be able to leave a comment on each question which would be very appreciated.

1. How satisfied are you with the information that SAGE finds, relevance of documents and how the information is then used in formulating the answer?
   - **Answer:**
   - **Comment:**
2. If you have experimented with different personalities, how satisfied are you with the changes in the behaviour and do you feel that they follow their instructions?
   - **Answer:**
   - **Comment:**
3. How satisfied are you with the application in general, primarily when interacting with it, sending messages, changing settings etc.?
   - **Answer:**
   - **Comment:**
4. If you have any other feedback or opinions on SAGE please let us know below. Could be related to anything from installation, use or SAGE's behaviour.
   - **Answer:**
   - **Comment:**

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

CHALMERS
UNIVERSITY OF TECHNOLOGY