



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

---

# **Empowering Automotive Software Development with LLM-RAG Integration**

A study on leveraging the RAG-framework for AUTOSAR and  
automotive safety standards and specifications

Master's thesis in Applied Data Science

**MIKAEL KIEU**  
**OSCAR BERGSTRAND**



MASTER'S THESIS 2024

# Empowering Automotive Software Development with LLM-RAG Integration

A study on leveraging the RAG-framework for AUTOSAR and  
automotive safety standards and specifications

MIKAEL KIEU  
OSCAR BERGSTRAND



UNIVERSITY OF  
GOTHENBURG

---



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering  
CHALMERS UNIVERSITY OF TECHNOLOGY  
UNIVERSITY OF GOTHENBURG  
Gothenburg, Sweden 2024

Empowering Automotive Software Development with LLM-RAG Integration  
A study on leveraging the RAG-framework for AUTOSAR and automotive safety  
standards and specifications  
MIKAEL KIEU OSCAR BERGSTRAND

© MIKAEL KIEU OSCAR BERGSTRAND, 2024.

Supervisor: Jonathan Thomas, Data Science and AI  
Advisor: Farshid Shafiabady, ZEEKR Technology Europe  
Examiner: Richard Johansson, Computer Science and Engineering

Master's Thesis 2024  
Department of Computer Science and Engineering  
Chalmers University of Technology and University of Gothenburg  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Typeset in L<sup>A</sup>T<sub>E</sub>X  
Gothenburg, Sweden 2024

Empowering Automotive Software Development with LLM-RAG Integration  
A study on leveraging the RAG-framework for AUTOSAR and automotive safety standards and specifications

MIKAEL KIEU

OSCAR BERGSTRAND

Department of Computer Science and Engineering  
Chalmers University of Technology and University of Gothenburg

## Abstract

The modern automotive industry increasingly relies on complex software systems such as Electronic Control Units that govern essential vehicle functions. The AUTOSAR standard provides a framework for automotive software development, ensuring interoperability, scalability, and compliance with safety regulations. However, adhering to AUTOSAR standards is challenging due to their complexity and the manual effort required.

This thesis investigates the potential of integrating the Retrieval Augmented Generation framework with Large Language Models to improve the explainability and usability of AUTOSAR specifications for software developers. The primary objectives are to develop a RAG-based model capable of interpreting queries about AUTOSAR specifications and generating clear, actionable steps for developers, and to evaluate the model's effectiveness in integrating retrieved context.

Our findings indicate that the RAG-framework with advanced techniques improves the contextual relevance and accuracy of generated outputs in the AUTOSAR domain. Specifically, the developed RAG model demonstrates higher levels of detail and precision but sometimes lacks fully actionable guidance. In comparison, GPT-4 and a naive RAG model, while generally accurate, often fail to provide the specificity required for complex software engineering tasks. Additionally, the study evaluates the efficiency of LLMs in synthesizing retrieved-context, noting significant improvements in newer models like GPT-4, while also recognizing ongoing challenges in consistently integrating complex information.

Limitations of this study include the focus on the AUTOSAR Classic Platform and the exclusion of graphical data. Future research should expand data extraction to include multi-modal inputs and explore fine-tuning and synthetic dataset generation to further improve model outputs. There is also potential for further research into context integration using more complex datasets to better understand the limitations of the model's capabilities.

Keywords: LLM, RAG, AUTOSAR, RAGAS, GPT4, GPT3, Yi, Context Integration.



## Acknowledgements

We would like to express our deepest gratitude to all those who have supported and helped us in this project.

Firstly, we would like to thank Jonathan for his unwavering guidance and insightful feedback. His dedication and willingness to help, often going above and beyond, have been invaluable to the success of our project. A pillar of support whom we could always rely on. With his support, we were able to explore new ideas and directions, leading to a richer and more comprehensive project. We are deeply grateful for his mentorship and the significant impact he has had on our academic journey.

We extend our sincere appreciation to Farshid, whose assistance, time, and engagement were crucial in making this project feasible. He provided a positive environment in which we could thrive, and his dedication and willingness to help have been invaluable. His commitment to our growth and development has left a lasting impact, and we are deeply grateful for his unwavering support.

We would also like to thank Richard for his understanding of our initial, sometimes confusing proposal. His constructive feedback guided this project towards a realistic and achievable goal.

Finally, we would like to express our heartfelt thanks to our loved ones for their unconditional love and support as we pursued our academic goals. Their encouragement has been our greatest strength throughout this journey.

Mikael Kieu, Oscar Bergstrand, Gothenburg, 2024-06-13



# Contents

<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xv</b>
<b>List of Acronyms</b>	<b>xvi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem description . . . . .	2
1.3 Objective . . . . .	2
1.4 Research Questions . . . . .	3
1.5 Contributions . . . . .	4
1.6 Limitations . . . . .	4
1.7 Related Work . . . . .	5
1.8 The outline . . . . .	7
<b>2 Theory</b>	<b>9</b>
2.1 Transformers . . . . .	9
2.1.1 Encoder and decoder . . . . .	9
2.1.2 Positional encoding . . . . .	10
2.1.3 Self-attention . . . . .	11
2.1.4 Multi-head attention . . . . .	11
2.2 GPT-Series . . . . .	11
2.2.1 GPT-3 . . . . .	12
2.2.2 GPT-4 . . . . .	12
2.2.2.1 Model training . . . . .	13
2.2.2.2 Performance . . . . .	13
2.3 Yi-Model . . . . .	13
2.4 Text and Sentence Embedding . . . . .	14
2.4.1 GTE Embedding Model . . . . .	15
2.5 Information Retrieval . . . . .	15
2.6 Retrieval-Augmented Generation . . . . .	16
2.6.1 Naive RAG . . . . .	17
2.6.2 Advanced RAG . . . . .	17
2.6.2.1 Pre-retrieval Strategies . . . . .	18
2.6.2.2 Retrieval Strategies . . . . .	20

2.6.2.3	Post-retrieval Strategies . . . . .	21
2.7	Prompt Engineering . . . . .	22
2.8	Automatic Evaluation . . . . .	22
2.9	Human Evaluation . . . . .	23
<b>3</b>	<b>Methodology</b>	<b>25</b>
3.1	The Experimental Procedure . . . . .	25
3.1.1	Connections to the Research Questions . . . . .	26
3.2	Data Collection . . . . .	26
3.3	The RAG Setup . . . . .	27
3.3.1	Data Pre-processing . . . . .	28
3.3.2	Embedding Model and Chunks . . . . .	29
3.3.3	Vector Database . . . . .	29
3.3.4	Retrieval . . . . .	29
3.3.4.1	Selection of Retrieval Technique . . . . .	30
3.3.4.2	Implementation of Retrieval . . . . .	31
3.3.5	Prompt Template . . . . .	32
3.3.6	Large Language Model . . . . .	32
3.4	Evaluation for The RAG Setup . . . . .	32
3.4.1	Human Evaluation - Survey . . . . .	33
3.4.2	Automatic Evaluation - RAGAS . . . . .	34
3.5	Context Integration Experiment . . . . .	34
3.5.1	Experimental Procedure . . . . .	35
3.5.2	Custom Dataset . . . . .	35
3.5.3	Integration Evaluation . . . . .	37
<b>4</b>	<b>Results</b>	<b>39</b>
4.1	Automatic Evaluation . . . . .	39
4.1.1	Faithfulness . . . . .	40
4.1.2	Answer Relevance . . . . .	41
4.1.3	Context Relevance . . . . .	43
4.2	Human Evaluation . . . . .	44
4.2.1	Does the response correctly address the question? . . . . .	44
4.2.2	How logically coherent do you find the reasoning in the response? . . . . .	45
4.2.3	Does the response contain any irrelevant information or inaccuracies? . . . . .	46
4.2.4	Which response do you find the most compelling, and which do you find the least compelling? . . . . .	48
4.3	Context Integration . . . . .	49
<b>5</b>	<b>Discussion</b>	<b>51</b>
5.1	From Generalist To Specialist . . . . .	51
5.2	RAGAS Under the Microscope . . . . .	52
5.3	Component Synergy Boosts RAG Performance . . . . .	53
5.4	Context Matters . . . . .	54
5.5	Ethical considerations . . . . .	55
5.6	Threats to Validity . . . . .	56

5.6.1	Internal Validity . . . . .	56
5.6.2	External Validity . . . . .	56
<b>6</b>	<b>Conclusion</b>	<b>57</b>
6.1	Future work . . . . .	58
	<b>Bibliography</b>	<b>59</b>
	<b>A Retrieval Prompt Template</b>	<b>I</b>
	<b>B Query Generation Prompt Template</b>	<b>III</b>
	<b>C Survey</b>	<b>V</b>
	<b>D Dataset Link</b>	<b>VII</b>



# List of Figures

1.1	Overview of the thesis structure. . . . .	7
2.1	Model architecture of the Transformer [19]. . . . .	10
2.2	Multi-Head Attention block illustrating several attention layers running in parallel [19]. . . . .	11
2.3	Embedding process for transforming words into numerical representation. . . . .	14
2.4	The RAG process from prompt to final output. . . . .	17
2.5	Illustration of a naive RAG model inspired by Gao et al. [13]. . . . .	18
2.6	Illustration of an advanced RAG model inspired by Gao et al. [13]. . . . .	19
3.1	Overview of the experimental procedure. . . . .	25
3.2	Foundational layers of AUTOSAR [5]. . . . .	27
3.3	Structured data before and after pre-processing. . . . .	28
3.4	Multi-index in the vector database. . . . .	31
4.1	Box plot presenting the distribution of FA scores for the naive RAG model. . . . .	40
4.2	Box plot presenting the distribution of FA scores for the developed RAG model. . . . .	41
4.3	Box plot presenting the distribution of AR scores for the naive RAG model. . . . .	42
4.4	Box plot presenting the distribution of AR scores for the developed RAG model. . . . .	42
4.5	Box plot presenting the distribution of CR scores for the naive RAG model. . . . .	43
4.6	Box plot presenting the distribution of CR scores for the developed RAG model. . . . .	44
4.7	Survey responses on correctly addressing the question Implementing CAN-FD. . . . .	45
4.8	Survey responses on reasoning for the question Implementing CAN-FD. . . . .	47
4.9	Survey responses on irrelevant information and inaccuracies for the question E2E Protect & Signal. . . . .	48
4.10	Pie chart presenting the compellingness of the responses. . . . .	49



# List of Tables

2.1	Summary of pre-retrieval methods. . . . .	19
2.2	Summary of retrieval techniques. . . . .	20
2.3	Summary of post-retrieval methods. . . . .	21
3.1	Methodology and experiment in relation to the research questions. . .	26
3.2	Comparison of the accuracy for different search algorithms. . . . .	30
3.3	Developed questions for evaluating the implemented models. . . . .	33
3.4	Comparison of the three different models. . . . .	33
3.5	Evaluative criteria and used survey questions. . . . .	34
3.6	Example questions and documents from the dataset for the context integration experiment. . . . .	36
4.1	Average RAGAS scores for the naive and developed RAG models. . .	39
4.2	Obtained results from context integration. . . . .	49



# List of Acronyms

**ECU** Electronic Control Unit

**LLM** Large Language Model

**RAG** Retrieval-Augmented Generation

**AUTomotive Open System ARchitecture** AUTOSAR

**AI** Artificial Intelligence

**NLP** Natural Language Processing

**CBR-RAG** Case-Based Reasoning for Retrieval Augmented Generation

**RLHF** Reinforcement Learning from Human Feedback

**SOTA** state-of-the-art

**IR** Information Retrieval

**QA** Question Answering

**RALM** Retrieval Augmented Language Models

**BOW** Bag of Words

**DPR** Dense Passage Retrieval

**FA** Faithfulness

**AR** Answer Relevance

**CR** Context Relevance

**CP** Classic Platform

**RTE** Runtime Environment

**BSW** Basic Software

**MTEB** Massive Text Embedding Benchmark

**Yi** Yi-34B-Chat-AWQ

**D1** Document 1

**D2** Document 2

- CT** Combination Text
- CN** Combination Numerical
- CXT** Contradiction Text
- CXN** Contradiction Numerical
- FI** Full Integration
- SDS** Single Document Synthesis

# 1

## Introduction

In this chapter background and description of the problem is provided. Subsequently, the research questions are presented, followed by the objectives this research aims to achieve and fulfill. Finally, the limitations and constraints impacting the research are addressed, followed by the report outline.

### 1.1 Background

The modern era in the automotive industry is heavily reliant on software, and vehicle systems are becoming increasingly complex. Software not only enhances adaptability and upgradeability without the need of hardware modifications, but it is also integral to both established and emerging vehicle functions. It supports essential systems, such as active safety, engine regulation, steering control, and infotainment [1]–[4], along with emerging functions such as autonomous driving.

The increasing dependence on software has introduced new risks and vulnerabilities, considering safety and security which in turn escalates the complexity of standardized vehicle systems [3]. The development of Electronic Control Unit (ECU), pivotal component in modern vehicles, exemplifies this increased complexity. ECUs are at the core of vehicle operations, from managing engine functions to ensuring passenger safety, underscoring the critical need for rigorous standards in software development to mitigate potential risks.

Efforts have been made to address the growing complexity within vehicle systems. One notable initiative is the global partnership between the automotive and the software industry, AUTomotive Open System ARchitecture (AUTOSAR) [5]. This partnership, which includes prominent companies such as BMW, Ford, Bosch and Continental, aims to establish a simplified framework for system architecture, development, and deployment of automotive software in a diverse set of platforms. The benefits of adhering AUTOSAR includes enhanced interoperability, improved safety and reduced development costs. By providing a standardized approach, AUTOSAR enables seamless integration of components from different suppliers, ensuring that they work together efficiently. Additionally, the framework supports scalability and flexibility, allowing manufacturers to adapt to evolving technological advancements and regulatory requirements. Nevertheless, adherence to AUTOSAR is a complex and challenging task as specialized knowledge and expertise are needed to navigate through the guidelines.

### 1.2 Problem description

The software automotive industry is a domain that is rapidly changing. Ever-evolving technologies give rise to new updates constantly, and the ambiguity of automotive standards poses additional challenges [6]–[8].

Research reveals significant challenges in the automotive software industry, especially with the extensive manual effort required to develop AUTOSAR-compliant code, as reflected in the paper by Smith and Khalid [9]. The authors proposed enhancing the workflow for developing AUTOSAR-compliant RTE code through the use of CAD tools to reduce the cost and time bottlenecks associated with manual coding. The paper also stresses scalability issues, noting that manually entered code must undergo extensive testing and verification against ISO standards.

The complexity of writing AUTOSAR-compliant code is also reflected in a survey conducted by Silver et al. [10]. The survey results indicate that while the most valued benefits of AUTOSAR include standardization, reuse, and interoperability, the predominant challenge remains in its complexity. In fact, 65% of respondents cited complexity as the primary drawback, alongside other notable concerns such as a steep learning curve and the significant initial investment in training personnel. The survey results also revealed a demand by the participants for a tool environment that enhances usability and called for more stable releases of AUTOSAR releases, which would help decrease the costs associated with migrating between versions.

Existing research has introduced AUTOSAR-based automation tools to address the challenges of AUTOSAR, focusing on software and generating code [9]. However, to the best of our knowledge, there have not been any studies on applying a Large Language Model (LLM) integrated with the Retrieval Augmented Generation (RAG) framework for enhancing the explainability of AUTOSAR specifications for software developers. The RAG-framework, which enriches LLMs by incorporating external knowledge sources, has been shown to reduce errors like hallucinations and reliance on outdated information in fields such as medical [11] and finance [12] domain. For AUTOSAR, applying the RAG-framework could provide up-to-date knowledge without the need for continuous model fine-tuning, thereby offering developers more accurate and actionable insights.

### 1.3 Objective

The primary objective of this thesis is to explore the integration of a RAG-framework with LLM within the AUTOSAR domain. The goal is to develop a RAG-based model capable of interpreting queries about AUTOSAR specifications and generating clear, actionable steps for software developers. This study aims to demonstrate how the application of advanced retrieval and generation techniques can significantly improve the ability of automotive software developers to adhere to AUTOSAR specifications efficiently.

## 1.4 Research Questions

The research questions of this study define the scope and serve as the core components for directing the methodology and analysis. The questions are outlined below:

**RQ1:** *How effective is the RAG-framework in providing contextual relevant outputs within the AUTOSAR domain?*

The RAG-framework has shown promise in various domains [11], [12], by enhancing the relevance and accuracy of generated outputs through integrating retrieved context.

In the highly specialized AUTOSAR domain, the ability to generate contextually relevant outputs can significantly impact the accuracy and utility of AI-driven solutions. Understanding the effectiveness of the RAG-framework ensures that the information generated aligns closely with the specific needs and standards of the domain, which is vital for making informed decisions in automotive software development.

Users and stakeholders in the automotive industry need to trust the outputs generated by AI systems. Demonstrating the RAG-framework’s ability to provide relevant and precise information can enhance this trust and encourage wider adoption of AI technologies within the industry. By rigorously evaluating the RAG-framework’s effectiveness with input from subject matter experts, we can assess its suitability for supporting decision-making processes and ultimately contribute to developing more effective and reliable AI-driven solutions in the AUTOSAR domain.

**RQ2:** *What impact do advanced techniques have on the outputs of the RAG-framework within the AUTOSAR domain?*

The RAG-framework has generated much interest in both academic and industrial literature and a range of improvements have been suggested. We experiment with a subset of advanced techniques covered in the works by [13], [14]. These techniques have the potential to increase the accuracy of the output and the reliability. Given the unique characteristics and requirements of the AUTOSAR domain, identifying the most effective techniques ensures that solutions are tailored to meet its specific needs, enhancing the overall utility of the RAG-framework.

A thorough investigation into the impacts of these techniques allows for the refinement of methods, ensuring that the solutions are robust and efficient. This process involves testing each component in isolation when possible or, if not, evaluating the quality of the overall output.

**RQ3:** *How efficiently do LLMs integrate and synthesize retrieved context in a RAG-framework setting?*

Regarding the RAG-framework, there is still limited knowledge about how information integration is executed. Our research question specifically probes the efficiency of LLMs in integrating and synthesizing retrieved context within an RAG-framework. The reliability of AI models in real-world applications hinges on their ability to consistently produce accurate outputs. Inconsistent or subpar integration of retrieved

context can lead to errors and reduce the trustworthiness of the system. Understanding how well LLMs perform this integration can inform the development of more robust and dependable AI systems.

## 1.5 Contributions

This thesis aims to contribute to the research on RAG-integrated LLM within the automotive industry. We will present the practical application of these technologies in the field and explore the necessary steps for a successful deployment. Inspired by our profound interest in the LLM domain, we hope this work foster further research and collaboration. We anticipate that our contributions will advance understanding and innovation in this area and benefit related domains by setting a foundation for future exploration.

The contributions of this theses are outlined as follows:

1. The AUTOSAR standard is critical to enhancing safety, standardization, and interoperability in automotive software engineering. Integrating the RAG-framework into this domain addresses the need for dynamic data handling and decision-making capabilities, essential for adapting to evolving industry standards and complex engineering challenges.
2. We provide valuable insights into an RAG model's ability to effectively provide relevant outputs based on specialized automotive specifications, paving the way for future research and improvements of RAG models. Current models often struggle with the complexity and specificity of automotive industry standards. By examining the performance of RAG models in interpreting and applying these standards, this contribution not only validates the model's effectiveness but also establishes a benchmark for further innovation and optimization in industry-specific applications.
3. We highlight potential challenges and limitations encountered in applying the RAG-framework, focusing on LLMs' ability to integrate and synthesize a retrieved context for generation. The reliability of a RAG model is dependent on its ability to consistently integrate and generate information. Any inconsistency or poor integration can lead to severe errors. By setting up an experiment with the focus point of context integration, we hope it will shed light and produce guidance towards the development of a more robust and dependent generative AI model.

## 1.6 Limitations

This thesis is limited to the software specifications of the AUTOSAR Classic Platform and the R23-11 release, no additional part of AUTOSAR is considered. This was agreed with subject matter expert and addressed a platform that they were interested in.

The pre-processing which includes both data extraction and data cleaning was limited in this project. As an extension of the data collection and preprocess would normally result in a time expensive process. This is usually considered to be a major bottleneck for deep learning projects [15]. The limitations entailed extracting only unstructured data in the form of free text and tabular data while excluding figures and other graphical-related data such as diagrams and charts.

These limitations suggests that the developed model is tailored specifically to the Classic Platform. As such, other platforms may exhibit unique characteristics and challenges that are not addressed by our study. Furthermore, the choice of excluding graphical content may affect the performance as the diagrams can contain key architectural insights and critical information for a full comprehension of AUTOSAR.

To address the shortcomings of excluding graphical data, there is potential of incorporating multi-modal models and advanced techniques for extracting and interpreting graphical content, such as deep learning methods for image captioning [16]. Enriching the database with more relevant context from the specification.

## 1.7 Related Work

The integration of LLMs has significantly progressed the field of Natural Language Processing (NLP). An area which focuses on enabling machines to understand, interpret, and generate human language [17]. Early models, such as n-gram models [18], primarily utilized probabilistic techniques to predict word sequences. At the same time, these models were limited in capturing long-range dependencies [18]. In contrast, contemporary LLMs, grounded in transformer architecture, have overcome these limitations and have markedly improved in generating contextually relevant text [19].

Despite the improvements with LLMs, they still face challenges when dealing with unseen data and often produce undesirable outputs related to biases [20] and toxicity [21]. Misleading content remains also a significant issue, as observed in works by [22]–[24], demonstrating that LLMs tend to generate incorrect information. This can undermine trust in LLMs, particularly in professional fields such as medicine [25] and law [26].

To mitigate the limitations that LLMs faces struggle in various downstream tasks, research has focused on improving the knowledge embedded within these models. A common strategy involves scaling up model parameters and increasing the training data, exemplified by the progression from LLaMA [27] to LLaMA2 [28] and from GPT-3 [29] to GPT-4 [30]. This approach is resource-intensive, necessitating substantial computational power and vast amounts of data[31].

An alternative approach requiring less resources is fine-tuning. Fine-tuning involves adapting a pre-trained model, which has already established general dependencies and relationships in language, to a specific domain using a smaller domain-specific dataset. Fine-tuning is particularly advantageous for tasks with scarce labeled data, as it enables the model to build on a strong foundational knowledge base rather than

learning from scratch. However, this method can become cumbersome if the dataset frequently updates, necessitating iterative model adjustments. There is also the risk of catastrophic forgetting, where the model loses its previously acquired knowledge during the fine-tuning process [32]. Additionally, fine-tuning may not be suitable when the required information is absent from the model’s pre-existing data [33].

A third promising approach to enhance LLMs’ knowledge is the RAG-framework. Unlike traditional methods that focus on scaling model parameters or fine-tuning, RAG-framework integrates external information retrieval mechanisms with generative capabilities. This hybrid solution addresses several limitations inherent in standalone LLMs, enhancing contextual awareness by grounding responses in specific documents or data sources.

Recent studies in the medical domain showcase the potential of RAG implementations to adhere to domain-specific instructions effectively [11], [34], [35]. Xiong et al. [11] conducted a large-scale experiment on the MIRAGE benchmark, which includes an extensive set of questions from five medical QA datasets. Their study demonstrated a relative performance increase using their RAG toolkit compared to chain-of-thought prompting [36]. Notably, GPT-3.5 and Mistral [37] integrated with a RAG-framework achieved comparable performance to GPT-4 [30] in terms of accuracy. Similar research in the legal domain has shown RAG’s effectiveness in improving the quality of generated output [38], [39]. One notable example is the CBR-RAG (Case-Based Reasoning for Retrieval Augmented Generation) model, which utilizes case-based reasoning to enhance the accuracy of legal question answering [40]. By retrieving relevant legal cases and statutes, the model provides a factually accurate context for its answers. This method significantly enhances the reliability of legal advice generated by the system, drawing on a vast repository of legal precedents and documents to support its responses.

Even with the numerous examples of RAG enhancing performance compared to standard LLMs, a recent study by Feldman and Shimei [41] has shown that RAG models can still be misled when prompts challenge the models’ pre-trained understanding. This study calls attention to the complex nature of hallucinations and errors that can occur even with accurate context. To overcome some of the challenges found in an RAG-framework, recent studies have seen extensive work bringing several strategies to improve its components. Examples include retrieval techniques to improve accuracy and relevancy of relevant documents [13], the use of query transformation [42] and the inclusion of an adaptive retrieval technique such as Self-RAG [43]. In the study by Yepes et al. [44], they explored an element-based chunking strategy in RAG models which involves taking into consideration the structure of the documents to guide the chunking process. Their findings suggested that this chunking strategy was able to improve RAG results on financial reporting.

While RAG models offer significant potential for enhancing the capabilities of LLMs, it is crucial to implement these systems with careful consideration of their limitations and challenges.

## 1.8 The outline

The outline of this report is presented in Figure 1.1, providing an overview of the structure of the thesis. Firstly, fundamental theory and research is presented and introduced in **Chapter 2**. This is followed by **Chapter 3**, describing the used method, covering details of the conducted research such as data collection and the experimental setup. In addition, it explains the steps taken to achieve the objectives of this research.

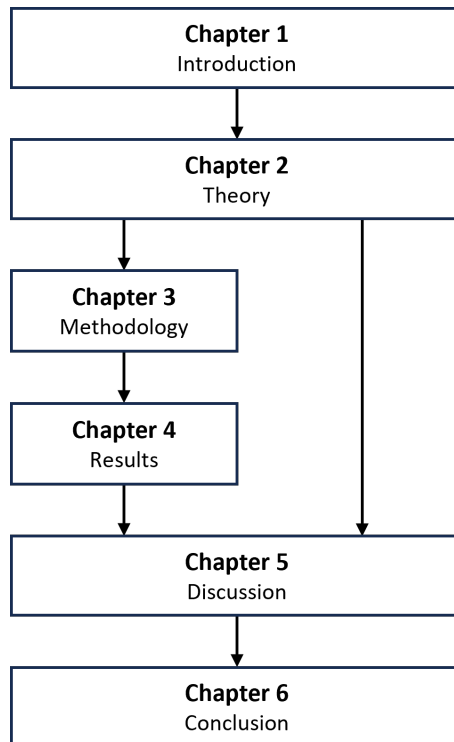


Figure 1.1: Overview of the thesis structure.

**Chapter 4** presents the result of this study, while **Chapter 5** provides an discussion of the obtained result. The discussion includes challenges, potentials, and limitations of the developed model. Thereafter, the study is concluded in **Chapter 6**, presenting closing statements and key takeaways.



# 2

## Theory

This chapter aims to present relevant and necessary theory to understand and provide a background of the topics being covered in this study. Specifically, the theory for transformers, large language models, embedding, retrieval information, retrieval-augmented-generation, prompt engineering, and evaluation methods are presented.

### 2.1 Transformers

The release of the transformers architecture by Vaswani et al. [19] significantly impacted the field of NLP and sequence modelling. It fundamentally changed the landscape of language tasks such as translation, summarization, and QA generation, giving rise to new standards for performance in deep learning.

The architecture is comprised of multiple stacked self-attention and fully-connected layers. These are employed in the encoder and decoder blocks, which can be found in Figure 2.1. Each layer has its distinct role in enhancing the model's proficiency for sequential understanding and generation.

The architecture incorporates several fundamental mechanisms, including positional encoding, self-attention, and multi-head attention. These components set it apart distinctly from earlier models like Recurrent Neural Networks [45] and Long Short-Term Memory Networks [46]. The role of self-attention is to create dependencies between the input data and output data, thus establishing connections between complex patterns over long sequences. This component overcomes the limitations of fixed-length contexts that exist in the traditional models. Moreover, a vital aspect of the transformer architecture is the capability of parallelization, which drastically reduces training time and enables efficient processing of large-scale datasets.

#### 2.1.1 Encoder and decoder

The encoder includes a stack of six identical layers, comprising two sub-layers, a multi-head self-attention mechanism, and a feed-forward fully connected layer. Additionally, each sub-layer incorporates a residual connection and a normalization layer to prevent vanishing gradient problems and reinforce the training process's stability as it maintains the input to a consistent scale.

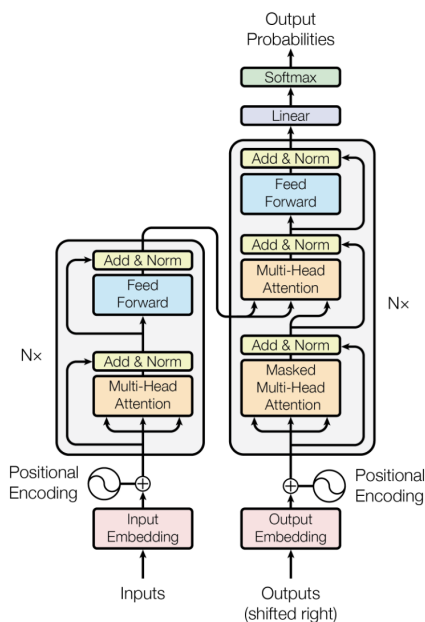


Figure 2.1: Model architecture of the Transformer [19].

Similarly, the decoder is composed of six identical layers. Each layer features the two sub-layers from the encoder and introduces a third sub-layer that performs multi-head attention over the output from the encoder stack. Each sub-layer is also equipped with residual connections and layer normalization.

The self-attention sub-layer in the decoder is also employed with a technique known as masked self-attention. This technique ensures that positions in the sequence do not consider future positions during the self-attention computation.

## 2.1.2 Positional encoding

The input data for transformers consists of word embeddings, which represent tokens in a  $d$ -dimensional space. In this space, tokens with similar semantic meanings are clustered closely together. However, these embeddings lack information about the relative positions of tokens within a sentence. This limitation is addressed by positional encoding, which accounts for the positions of tokens in a sequence. Positional encoding allows the model to consider both the position and the meaning of tokens, which is crucial for understanding language.

The equation to compute the positional encoding are shown in Equations 2.1 and 2.2, for which  $pos$  denotes the position and  $i$  denotes the dimension.

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{(2i/d_{\text{model}})}}\right) \quad (2.1)$$

$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{(2i/d_{\text{model}})}}\right) \quad (2.2)$$

### 2.1.3 Self-attention

Self-attention layers are instrumental in capturing dependencies within a sequence. By utilizing self-attention, each element in the input is compared to all other elements, generating a dynamic representation. This feature allows the model to understand different parts of the sequence simultaneously. The attention function, as presented in Equation 2.3, is computed on a set of queries (Q), keys (K), and values (V), all of which are stored in their respective matrices. First, the dot product of the queries and keys is computed. Afterwards, scaling is applied, and the softmax function is used to compute the attention scores.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.3)$$

### 2.1.4 Multi-head attention

Multi-head attention refers to the leveraging multiple attention mechanism operating in parallel. This enables the model to concentrate on various features within the input sequence at the same time. Specifically, this is accomplished by organizing the input queries, keys, and values into multiple "heads," allowing the model to attend to different representations in the subspaces concurrently. An illustration of this idea is shown in Figure 2.2. Each head individually applies the attention function to its own set of projected queries, keys, and values, resulting in a set of output values. These outputs are then concatenated and passed through a final linear projection to produce the combined output.

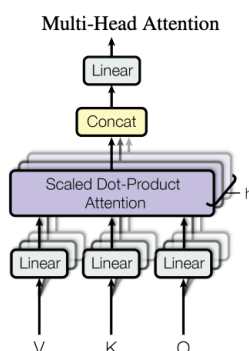


Figure 2.2: Multi-Head Attention block illustrating several attention layers running in parallel [19].

## 2.2 GPT-Series

In this section the GPT-Series including GPT-3 and GPT-4 are presented. This by covering details of model training and performances.

### 2.2.1 GPT-3

GPT-3, introduced by Brown et al. [29], is an autoregressive language model which predicts future values based on past values. It features 175 billion parameters, making it ten times larger than any preceding non-sparse model. This model expands upon the concept by Radford et al. [47] of using a pre-trained language model capable of interpreting instructions provided at inference. The rationale for increasing the model's parameters is rooted in the observation that enhanced in-context learning capabilities correlate positively with scale. Although architecturally similar to its predecessors, GPT-3 distinguishes itself through a significant increase in model size and training data.

In conjunction with GPT-3, a modified version of it was released named GPT-3.5. This also came with its own variants, such as GPT-3.5-turbo. The architectural design and parameter size are undisclosed, but it is also optimized for end users, offering faster performance and reduced cost for token usage.

Besides being trained on an extensive corpus of data, the GPT-3 model was not explicitly fine-tuned for a specific task. Instead, the task is determined during inference through the use of prompts. This method conditions the model to respond to instructions or examples of the desired task. The model is then expected to complete the task by predicting subsequent words based on the context. This training methodology mirrors the human cognitive process of creating new language tasks from a few examples, a challenge that earlier NLP systems struggled to overcome.

The model underwent three different learning modalities to enable flexibility to different requirements, as described by Brown et al [29].

- Few-shot learning: GPT-3 uses multiple prompt examples to guide its responses.
- One-shot learning: The model infers task requirements from a single example.
- Zero-shot learning: GPT-3 relies solely on the natural language instructions within the prompt.

At the time of release, the performance evaluations across various benchmarks showed GPT-3 matching or surpassing many specialized models, highlighting its robustness and versatility. The model's performance popularized the focus on developing general-purpose LLMs, establishing GPT-3 as a milestone in practical, versatile LLM applications in AI.

### 2.2.2 GPT-4

The model GPT-4, released by the OpenAI team [30], is a transformer-based model pre-trained on a vast corpus of text data from publicly available sources and data licensed from third parties. While the exact size of GPT-4 in terms of the number of parameters is not disclosed, it follows the trend of scaling up from previous versions, suggesting a substantial increase in size and complexity compared to GPT-3 [29]. The capacity to process information was expanded from a maximum of 4096 tokens in the previous model to 8192 tokens in the current version.

### 2.2.2.1 Model training

GPT-4’s training process can be divided into two main phases: pre-training and fine-tuning. The pre-training was done using a diverse dataset comprising publicly available data and data licensed from third-party providers. After pre-training, GPT-4 underwent fine-tuning using Reinforcement Learning from Human Feedback (RLHF). This process involves human evaluators providing feedback on a model’s outputs, which is then used to adjust the model’s parameters to better align with human judgment and preferences, detailed by Ouyang et al. [48]. RLHF helps improve the model’s performance on specific tasks and enhances its safety by reducing undesirable outputs, such as biased or inaccurate information.

A notable aspect of GPT-4’s development was the emphasis on predictable scaling. The OpenAI team developed infrastructure and optimization methods that behave predictably across various scales. This approach allowed for reliable predictions about GPT-4’s performance based on smaller models trained with significantly less computational power.

### 2.2.2.2 Performance

GPT-4 demonstrates a remarkable ability across numerous languages, outperforming existing large language models and state-of-the-art (SOTA) systems on a collection of NLP tasks. The evaluation of GPT-4 extends to simulations of exams designed for humans, where it exhibits exceptional performance, even ranking in the top 10% for challenging tests such as the simulated Uniform Bar Examination. This accomplishment highlights the model’s ability to process and generate language and its potential for application in contexts requiring advanced reasoning and knowledge [30].

Despite the progress made with GPT-4, it still shares limitations with earlier GPT models, such as proneness to generating misleading information and making reasoning errors. These limitations necessitate cautious application, especially in high-stakes scenarios, underscoring the importance of ongoing efforts to improve model safety and reliability [30].

## 2.3 Yi-Model

The 01.AI research team introduced the Yi-series, bilingual language models trained on a robust 3T multilingual corpus [49]. The Yi-series includes 6 billion and 34 billion parameter models, each designed to balance scale, data volume, and quality effectively. Notably, the 34-billion-parameter model is optimized for efficiency, performing complex reasoning tasks on consumer-grade hardware such as the NVIDIA RTX 4090. Although smaller than some competitors like LLaMA [27], the Yi-series compensates with enhanced training data volumes.

Built on the transformer framework, the Yi-models incorporate elements from the LLaMA architecture, such as the Grouped Query Attention [27] mechanism. This feature, applied in both the 6B and 34B models, groups query heads to share key and value heads, reducing costs without compromising performance. The models

applies the SwiGLU activation function to reduce the activation size and maintain consistency with the standard post-attention layer [50]. It involves taking the product of two linear transformations, with one being modified by another activation function, the sigmoid function. The design helps to maintain parameter efficiency, crucial for managing the streamlined parameter count.

Additionally, Rotary Positional Embedding [51] has been adapted to support up to 200,000-word contexts, initially training on 4,000-word sequences and scaling to accommodate longer texts. This feature underpins the models' ability to handle extensive data sets effectively.

Data quality was a priority in the Yi series development, with a comprehensive pre-training that includes several layers of filtering—ranging from language filter, text quality, semantic filtering and safety measures.

Upon release, the Yi-34B model has outperformed all existing open-source models on various benchmarks such as MMLU, BBH, CMMLU, GSM8s and C-Eval, it does not surpass GPT-4 in any of the task-specific benchmarks, such as MATH [49].

## 2.4 Text and Sentence Embedding

Embedding is a technique that transforms words into numerical representation. This approach is crucial for various NLP tasks, as it aids the representation of textual content in a more computationally manageable form. Traditional methods like TF-IDF [52] uses sparse representations based on a vocabulary derived from document corpora. However, these techniques struggle with words that have multiple meanings depending on the context. In response, sentence embedding models, also referred to as dense representation, have been developed to capture the entire sentence in a single vector, thereby preserving the semantic meaning of the text. This method can significantly improve the performance for downstream tasks in complex NLP applications as better meaning of the text is captured with a reduced dimensionality [53]–[55].

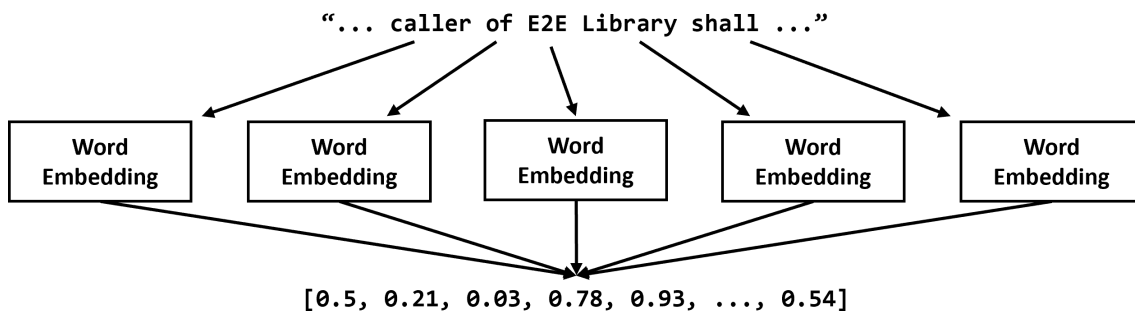


Figure 2.3: Embedding process for transforming words into numerical representation.

### 2.4.1 GTE Embedding Model

The GTE embedding model introduced by Li et al. [53] is a general purpose text embedding model that is trained with multi-stage contrastive learning to create high-quality text embeddings.

The primary goal of contrastive learning is to distinguish between similar and dissimilar pairs of data points. For text embeddings, this involves making the representations of semantically similar texts (positive pairs) close to each other in the embedding space, while pushing the representations of dissimilar texts (negative pairs) farther apart.

The GTE model employs an improved contrastive loss that goes beyond the standard in-batch negatives approach. The improved loss function considers both query-to-document and document-to-query directions. It includes negative samples from both queries and documents within the batch, effectively enlarging the negative sample pool. Given a batch of positive text pairs  $B = \{(q_1, d_1), (q_2, d_2), \dots, (q_n, d_n)\}$ , the improved contrastive loss is formulated as:

$$L = -\frac{1}{n} \sum_{i=1}^n \log \frac{e^{s(q_i, d_i)/\tau}}{Z} \quad (2.4)$$

where the partition function  $Z$  is:

$$Z = \sum_j e^{s(q_i, d_j)/\tau} + \sum_{j \neq i} e^{s(q_i, q_j)/\tau} + \sum_j e^{s(q_j, d_i)/\tau} + \sum_{j \neq i} e^{s(d_j, d_i)/\tau} \quad (2.5)$$

The training process for the model is split into unsupervised and supervised phase. In the unsupervised phase, the model is trained on a large dataset of text pairs extracted from various sources without human labels such as CommonCrawl dataset [56]. The text pairs can be in the form of questions and answers on forums or linked documents on web pages. The objective is to learn general text representations that capture a wide range of semantic relationships. In the supervised phase, the model is fine-tuned on annotated text triples, where each triple consists of a query, a positive document, and a negative document. The fine-tuning data includes high-quality, human-labeled pairs from various tasks like web search, question answering, and natural language inference. This stage refines the embeddings to be more accurate and task-specific.

## 2.5 Information Retrieval

Information Retrieval (IR) is defined as a method of seeking out relevant information from vast collections of unstructured or semi-structured data, predominantly text-based, according to Manning et al. [57]. In today’s data-driven and information-rich world, the application of IR is applied in various practical settings, such as web searches, question answering (QA), personal assistants, and chatbots [58].

The task of IR can be divided into two sub-tasks, retrieval and ranking. Retrieval can be explained as the task of finding and locating relevant information, while ranking is used to rank the relevant information as multiple documents may be relevant for a specific query. The traditional way matched terms from the query with the documents and thereby located related information. However, recent progress in NLP have greatly impacted IR and now utilize deep learning techniques to address the limitations of matching terms, such as synonyms and lexical in-alignments [58]. The most commonly recognized types of IR models are boolean, vector space model, and probabilistic models [57].

Boolean retrieval is referred to as the traditional way of retrieval and is term-based. This by utilizing boolean logic to match words in queries to documents, providing results that exactly match the words specified in the query [59]. Hambarde and Proenca [58] describe boolean retrieval as simple, fast, and straightforward but insufficient in terms of managing scenarios of synonyms and contextual nuances.

Retrieval done by a vector space model is solely phrased-based by representing documents and queries as vectors in multidimensional space. This is done by utilizing the capabilities of a language model to incorporate a weighting scheme for the used terms. Moreover, the vector space model is a widely adopted model for the task of IR and retrieves relevant information based on vector similarity [58]. Metrics such as cosine similarity [60] and euclidean distance [61] determine the similarity between documents and queries.

The probabilistic model is described as an extension of the vector space model, as it also incorporates weighting terms and their dependencies [58]. Additionally, the probabilistic model estimates the probability that a given document will be relevant to a user’s query, based on the likelihood of occurrence of terms within the document [62].

## 2.6 Retrieval-Augmented Generation

The concept of RAG was first introduced by Lewis et al. [63]. This innovative framework represents a breakthrough in NLP, combining the strengths of parametric models, such as pre-trained seq2seq models, with non-parametric systems like external databases to enhance the quality of text generation. The core principle of this approach is that by integrating external information, it will reduce the inherent limitations of the language models such as the tendency to generate hallucinated (incorrect or misleading) content or outdated information [64]. The authors discovered that for language generation tasks, their RAG model outperformed parametric-only models such as BART in the task of open domain Q&A. As it produced more factual and specific responses, than its counterpart.

Expanding on the RAG-framework, Ram et al. [65] introduced a similar approach called Retrieval Augmented Language Models (RALM). This approach differs slightly from the RAG-framework introduced by Lewis et al. [63], as it instead employed a language model such as GPT-2 as the generation component instead of a seq2seq model. Ram et al. also proposed maintaining the language model architecture intact

and prepending grounding documents to the input. Their research demonstrated that off-the-shelf general retrievers significantly enhance the performance of language models across various sizes and diverse corpora in terms of perplexity. The findings suggest that RALM offers substantial potential to enhance the grounding of language models, especially in scenarios where a pre-trained language model must be deployed without modifications. Figure 2.4 illustrates a RAG pipeline, showcasing its process of integrating external data for improved output with factual knowledge [63].

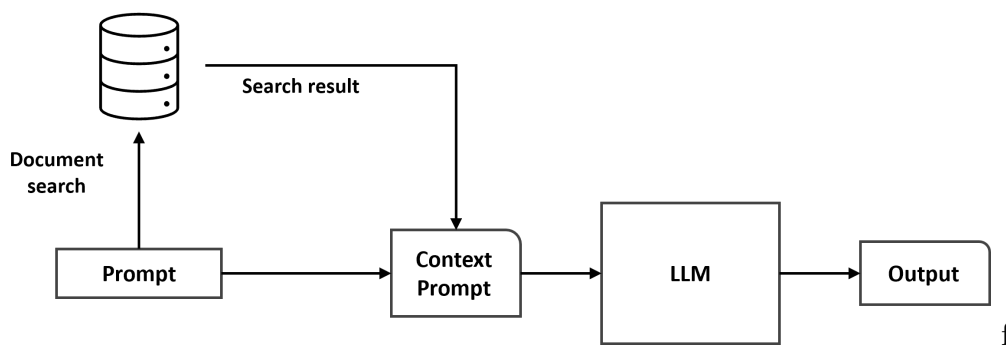


Figure 2.4: The RAG process from prompt to final output.

Substantial progress have been made in developing the RAG-framework, as documented in a survey by Gao et al.[13]. The survey categorizes RAG into three categories: Naive, Advanced, and Modular RAG. The details of Naive and Advanced RAG will be explored in-depth in Section 2.6.1 and 2.6.2 respectively.

### 2.6.1 Naive RAG

A Naive RAG model consists of three components: indexing, retrieval, and generation, as depicted in Figure 2.5. The indexing phase involves data preparation, including cleaning and extraction from various file formats to eliminate noise and irrelevant information. To fit within the language models context window, the data is divided into manageable segment, also referred to as chunks. Next, the chunks are transformed into embeddings with an embedding model and stored in a vector database [13].

In the retrieval component, a user’s query is embedded and compared with vectors in the external database. The system retrieves data segments that best match the query, often quantifying the semantic similarity with a cosine similarity. These segments provide a richer context to the LLM, enhancing its response accuracy.

The generation component combines the query and retrieved content into a single prompt for the LLM. This enables the generation of responses that are both contextually enriched and task-specific. The scope of LLM responses is confined to the provided prompt information.

### 2.6.2 Advanced RAG

The Advanced RAG was developed to address the limitations of the naive RAG, such as the retrieval quality and inefficiency in managing complex queries [13]. Advanced

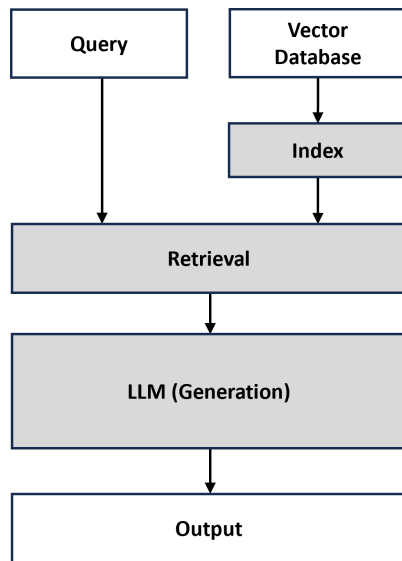


Figure 2.5: Illustration of a naive RAG model inspired by Gao et al. [13].

RAG can be described as an enhancement by implementing several strategies under the categories of pre-retrieval and post-retrieval. The steps of advanced RAG are presented in figure 2.6, a simplified model indicating where these strategies take place.

The process of advanced RAG is similar to what is described in section 2.6.1, with the exception of pre-retrieval and post-retrieval strategies. Pre-retrieval strategies refer to activities that refine the data quality to obtain a greater index. For instance, improvement of text and content quality, integrating metadata and ensuring data consistency and relevance, which all relate to preparing the data for efficient retrieval [13].

Moreover, re-ranking, techniques to ensure data diversity and relevance, and prompt compression are examples of post-retrieval strategies. To directly pass the retrieved content into the context window of LLMs can lead to, for example, information overload. The goal of post-retrieval strategies is to enhance data granularity by minimize noise from the retrieved content, particularly important when the content is extensive [13].

The goal of post-retrieval strategies is to enhance data granularity by minimizing noise from the retrieved content. This results in providing LLMs with essential data in a well-structured manner, which is particularly important when the context is extensive.

### 2.6.2.1 Pre-retrieval Strategies

According to Gao et al. [13], pre-retrieval covers techniques to optimize indexing structures and refine the original query to enhance index quality, which is crucial for accurate context retrieval during the retrieval phase. In Table 2.1 four such methods are presented. Chunk optimization, index structure, adding metadata, and query alignment all possess the ultimate goal of enhancing data quality and data

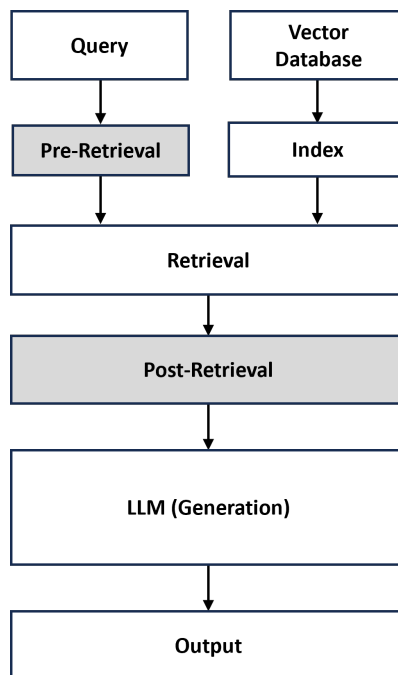


Figure 2.6: Illustration of an advanced RAG model inspired by Gao et al. [13].

granularity. This by preparing it for efficient retrieval. Bad content quality most likely leads to hallucinations done by the model as observed by Zhao et al. [14].

Table 2.1: Summary of pre-retrieval methods.

Method	Description
Chunk Optimization	Activity to optimize the chunks
Index Structure	Process to create structure with multiple index paths.
Adding Metadata	Incorporate metadata such as file names and keywords to improve retrieval.
Query Alignment	Modify the input query to address document alignment disparities and retrieve a richer context.

Chunking strategy revolves on dividing documents into fixed token lengths, such as 100, 256, or 512 tokens [66]. Larger chunks capture more contextual information but may include irrelevant details, increasing processing time and costs. Conversely, smaller chunks reduce noise but might miss necessary context, prompting the use of recursive splits and sliding window methods for more effective retrieval [13].

Hierarchical index structures in RAG models facilitate efficient data retrieval. This structure can include parent-child relationships among chunks or data summaries at each node to expedite traversal. Alternatively, a knowledge graph index can be employed to map connections between concepts and entities [67]. This benefits the query routing technique, a strategy where the query is routed in specific indexes to explore several sources in a variety of formats and domains [13].

Adding metadata is summarized as method to enhance the retrieval [68] performance. This method can be done in different ways, for instance, by adding file names, section names, chapters, and keywords to mention a few. This method is in place to filter retrieved documents, and support in extracting required information [13], [14], [68].

Query Alignment or Query Expansion refers to breaking down queries to yield a larger number of documents related to the original query. It involves modifying the input query to retrieve more rich relevant context and accurate retrieval results. This aids in addressing alignment disparities in the index structure [13], [14], [69].

### 2.6.2.2 Retrieval Strategies

There are two common categories for retrieval, sparse and dense passage retrieval. These can be used independently or in a combination of one and another. In Table 2.2 three common retrieval techniques are presented, which are keyword-based search, vector search, and hybrid search.

Table 2.2: Summary of retrieval techniques.

Method	Description
Keyword-based Search	Sparse retrieval technique, ranking documents by the frequency of terms, for instance, the BM25 algorithm.
Vector Search	Dense retrieval technique, documents are represented as vectors and retrieved by semantic similarity.
Hybrid Search	Combining the sparse and dense retrieval techniques.

Keyword-based search is a sparse retrieval technique, utilizing the frequency of words in a document for the retrieval process. BM25 is an example of keyword-based search, providing accurate and relevant documents based on ranking the term frequencies which is based on the work done by Robertson and Jones [70]. Sparse retrieval relies on high-dimensional sparse representations of documents and methods such as Bag of Words (BOW) and TF-IDF. This technique is especially effective for retrieving documents that precisely match specific terms, additionally, is significantly cheaper than neural alternatives such as dense passage retrieval [65]. Additionally, Ram et al. [65] stated that BM25 outperforms dense retrievers, which is consistent with previous work across several tasks.

Karpukhin et al. [55] introduced Dense Passage Retrieval (DPR) for open-domain question answering. Vector search, as described in Table 2.2, falls under the category of DPR. This technique utilizes a neural network to represent documents as dense vectors, allowing the retrieval of the top-k documents based on their semantic relationship to the query.

The hybrid search consists of combining a neural network and the keyword-based algorithm to improve the accuracy and relevancy of the search results. As this combined search algorithm aims to identify various relevance features and can mutually benefit by utilizing complementary relevance information. This approach was able to boost the performance for a frozen LLM in the study by Levine et al. [71].

### 2.6.2.3 Post-retrieval Strategies

Post-retrieval strategies are applied after document retrieval to augment retrieved documents in conjunction with the query for input to an LLM. These strategies take into account the context window of LLMs and process the retrieved context to enable LLMs to capture essential information while minimizing noise [13]. In Table 2.3, two post-retrieval methods are presented, Re-Ranking and Information Compression, both designed to enhance the quality and alignment of the retrieval result.

Table 2.3: Summary of post-retrieval methods.

Method	Description
Re-ranking	Considers the context window and rearrange the order of documents.
Information Compression	Managing large-scale information, by improving quality and alignment.

Re-ranking addresses the issue of LLMs commonly declining performance when additional context is presented as input. Re-ranking models are used to enhance LLMs focus on the most relevant chunks, such models reorder the retrieved context for diversity and better results. For instance, this can be done by relocating important chunks to the edges of the prompt [13], [14]. Barnett et al. [68] note the importance of retrieving relevant documents. As they identified a prevalent failure point in RAG models, although the answer to a question may be present within a document, it is often not ranked highly enough to be returned to the user.

Gao et al. [13] discuss the capability of information compression when managing significant amounts of information both in retrieval and prompting LLMs. Information compression is crucial when handling large volumes of documents, as an abundance of information can introduce noise and diminish the LLMs' effectiveness in identifying relevant content. A practical approach involves utilizing the LLM to assess the retrieved documents, enabling it to discard those with low relevance. This selective filtering ensures that only the most pertinent information is retained, which improves the LLM's ability to focus on and process the essential content efficiently.

Pawar et al. [72] mention that LLMs face limitations considering the context length, while Zhang et al. [73] state that tasks such as RAG require LLMs to process long contexts. However, LLMs have different context lengths, and these context lengths do not determine the optimal number of documents to return to the LLMs. The early stages of research suggested that providing more documents improved outputs, while others stated that only providing a selected set of documents outperforms full context length. This could be due to factors such as increased relevance and quality of the context [74]. However, Liu et al. [75] found that more than 20 retrieved documents marginally improved the accuracy of GPT-3.5-Turbo. Hsia et al. [74] found that encoder-decoder models can more effectively than decoder-only process tens of documents.

## 2.7 Prompt Engineering

The capabilities of LLMs can be significantly enhanced through the introduction of prompt templates that contain task-specific instructions, directives, or guidance. This practice, widely recognized as prompt engineering [76], has proven to be an effective mechanism for fine-tuning the outputs of LLMs [77]. Amatriain [78] further elaborates on the significance of prompt templates within the context of generative AI models as inputs that steer the model’s outputs and provide a reproducible method of interaction with an AI.

Prompt engineering encompasses various techniques in the form of zero-shot learning [79], few-shot learning [80], and chain-of-thought processes [36]. While Amatriain also mentions subtle yet significant details that can emphasize instructions within a prompt template, such as the use of capital letters or exclamation marks

In a RAG-framework setting, the context retrieved is seamlessly integrated into the original prompt for the LLM, creating what can be described as an augmented prompt. This enhancement not only boosts the generative capabilities of LLMs but also enables more accurate and fact-based responses [77]. Barnett et al.[68] note that RAG models particularly require customized prompts, especially in question-and-answer scenarios and for specific formatting instructions.

## 2.8 Automatic Evaluation

Es et al. introduced Retrieval Augmented Generation Assessment (RAGAS), a comprehensive evaluation framework for RAG pipelines [81]. RAGAS offers a suite of metrics to evaluate various aspects of RAG models, both with and without reliance on ground truth annotations. Among these metrics, Faithfulness (FA), Answer Relevance (AR), and Context Relevance (CR) stand out as they do not require ground truth annotations. Additionally, these metrics was constructed by leveraging a LLM, the referenced paper utilized GPT 3.5 - Turbo, whereas this study leveraged GPT-4.

### Faithfulness

FA measures if the model is basing its output on the retrieved context, in other words the models inclination to hallucinate. FA is computed by prompting a LLM to deconstruct the generated output into a set of discrete statements. These statements are then being fed to another LLM to verify if the statements can be inferred from the retrieved context. For each statement, a yes or no verdict is determined by the LLM and the overall ratio between supported statements and the total number of statements reflects the FA score. As shown in Equation 2.6:

Faithfulness ( $FA$ ) is defined as the ratio of supported statements  $V$  and the total number of statements  $S$  and the score is a scale between 0 and 1.

$$FA = \frac{|V|}{|S|} \quad (2.6)$$

### Answer Relevance

AR measures if the generated output is relevant to the query. Relevant in this term pertains to if the generated answer addresses the query in a complete or if it contains redundant information. Another aspect to consider with this metric is that it does not consider the factual correctness in its evaluation.

This metrics operates by prompting a LLM to generate n amount of potential questions based on the generated answer. Subsequently, these questions are transformed into vector representations using an embedding model. In our study, the GTE embedding model was implemented to transform these questions into a vector representation. The embeddings of the newly generated questions are then compared to the embeddings of the original query. This comparison is computed with a cosine similarity [60] score. The answer relevance metric is computed as shown in the Equation 2.7 where:

$AR$  is computed as the average of the similarity measures between a query  $q$  and a set of potential queries  $q_i$ , where  $i$  ranges from 1 to  $n$ . The score is given in a scale between 0 and 1.

$$AR = \frac{1}{n} \sum_{i=1}^n \text{similarity}(q, q_i) \quad (2.7)$$

### Context Relevance

The CR metric measures if the provided context is relevant if it exclusively contains informations that is needed to answer the provided query. This metric has a penalty factor by taking into consideration if the retrieved context includes redundant information. To compute CR, an LLM is prompted to identify and extract sentences that are directly relevant to answering the provided query. The CR score is then computed by determining the proportion of these extracted sentences relative to the total number of sentences within the context. The CR is defined in Equation 2.8 as the ratio of the number of extracted sentences to the total number of sentences in the context. The score is given in a scale between 0 and 1.

$$CR = \frac{\text{number of extracted sentences}}{\text{total number of sentences in context}} \quad (2.8)$$

## 2.9 Human Evaluation

The significance of automatic metrics in providing quantitative evaluations of generated text quality cannot be overstated. However, relying exclusively on these metrics can be insufficient, as mentioned by Celikyilmaz et al. [82]. The authors assert that

human evaluators play a crucial role and are often considered gold standard when evaluating outputs of LLMs. Automatic metrics have also been reported to pose risks of factual inaccuracies, which human evaluation can help mitigate [83].

Ideally, human evaluation should complement automatic metrics to ensure a comprehensive assessment of text-generated outputs. The importance of human evaluators is particularly important in specialized fields such as healthcare [84] or medical field [85], where the reliability of the information is critical. This is also reflected in AUTOSAR domain as safety concerns are paramount as it deals with automotive software architecture where errors can lead to severe malfunctions.

Despite its advantages, human evaluation introduces its own set of challenges. It significantly increases the cost and time required for the evaluation process, potentially limiting its feasibility. Moreover, human evaluation does not always guarantee standard procedures or reproducibility, leading to variability in results.

For effective human evaluation, several best practices should be considered. It is crucial to define the criteria relevant to the specific task at hand. Common criteria include correctness, reasoning, potential inaccuracies, and the logical progression of the output [85], [86]. Choosing proper evaluators is crucial, as they must accurately reflect the intended audience of the generated text. It's important to recognize that the preferences and biases of evaluators can impact their assessments. Thus, assembling a diverse and representative group of evaluators is essential to achieve balanced and impartial evaluation outcomes.

# 3

## Methodology

This chapter outlines the used methodology. Firstly, an overview of the experimental procedure and its connection to the research questions is provided. Thereafter, the scope of the data collection and the used data are presented. This is followed by the RAG set-up, including pre-processing, retrieval, and evaluation. Finally, the chapter addresses the intricacies of context integration.

### 3.1 The Experimental Procedure

The developed RAG model is based on the theory discussed in **Chapter 2**. This experiment and model consists of several steps and components, illustrated in Figure 3.1, which outlines the experimental flow and provides an overview of the model's architecture. **Section 3.2** details the first step, focusing on the used and collected data from AUTOSAR Classic Platform. The data is central to the model's ability to retrieve and generate information, entailed through pre-processing described in **Section 3.3.1** and **Section 3.3.2**, which covers the embedding model and chunking strategy.

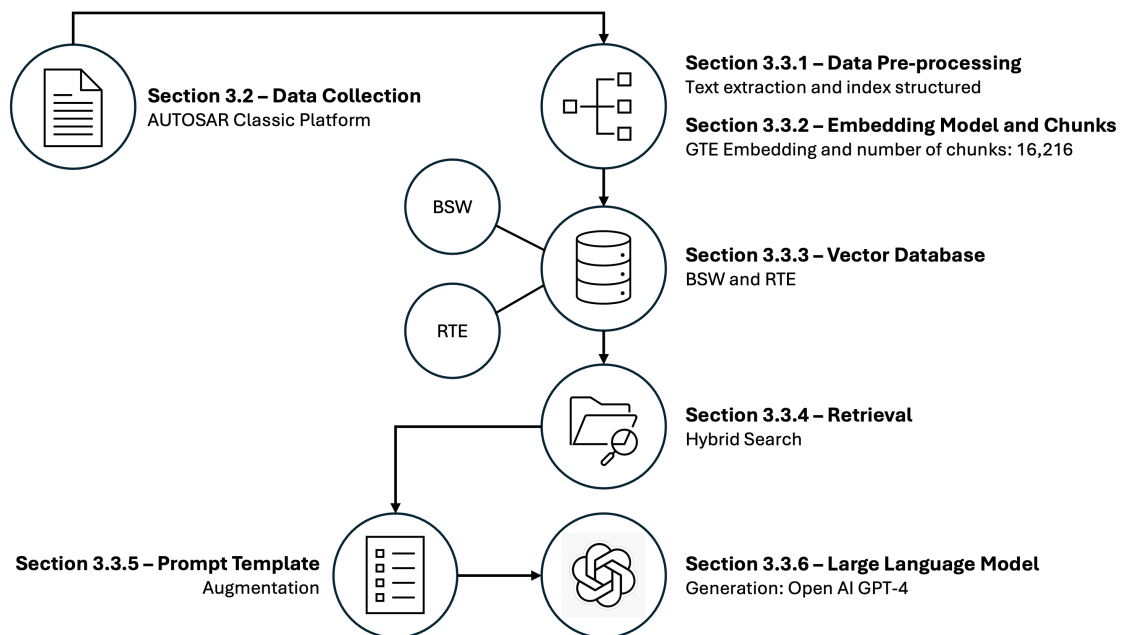


Figure 3.1: Overview of the experimental procedure.

The following step in this experiment involved the vector database, a critical component detailed in **Section 3.3.3**. Weaviate was used for efficient data management and storage of high-dimensional data. **Section 3.3.4** describes the retriever in-depth, explaining its role in the vector database and how the retrieval technique BM25 was selected and implemented. This is followed by **Section 3.3.5**, which discusses how the data was augmented and the prompt template crafted and tuned. Finally, the use of LLMs is presented, covering motivation and integration aspects in Section 3.3.6.

### 3.1.1 Connections to the Research Questions

The primary objective of our methodology is to address the research questions summarized in Table 3.1. The first two research questions focus on leveraging the RAG-framework and advanced techniques within the AUTOSAR domain. The third research question explores the efficiency of LLMs in integrating and synthesizing contexts, crucial for understanding the utility and adaptability of the RAG-framework.

Table 3.1: Methodology and experiment in relation to the research questions.

ID	Research Question	Method Component
1	How effective is the RAG-framework in providing contextual relevant outputs within the AUTOSAR domain?	Theoretical Framework Data Collection Experiment Analysis
2	What impact do advanced techniques of the RAG-framework have on the outputs within the AUTOSAR domain?	Theoretical Framework Data Collection Experiment Analysis
3	How efficiently do LLMs integrate and synthesize retrieved context in a RAG framework setting?	Experiment Analysis

## 3.2 Data Collection

The purpose of the data collection was to collect a comprehensive set of AUTOSAR specifications, allowing accurate answers to questions of interest. The collected data covered the software specifications of the Classic Platform (CP) released in November 2023, in total 128 documents [5].

AUTOSAR is a collective effort by the automotive industry to establish a simplified and standardized framework for system architecture, development, and deployment of automotive software across multiple platforms. The complete AUTOSAR CP consists of 229 documents including requirements, templates, software specifications, explanations, models, and technical reports. The AUTOSAR CP software architecture is three-layered and runs on a microcontroller, which is illustrated in Figure 3.2.

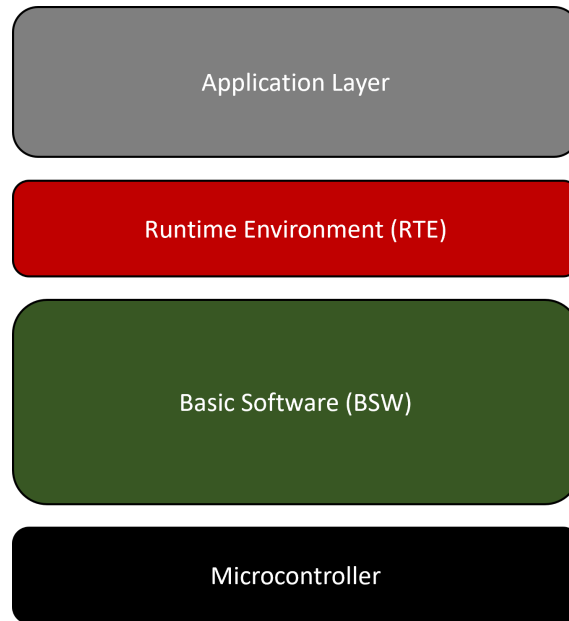


Figure 3.2: Foundational layers of AUTOSAR [5].

The top layer of AUTOSAR CP is the Application Layer, which covers aspects of modular software development through AUTOSAR-defined interfaces. The second layer, Runtime Environment (RTE), is the core of AUTOSAR and acts as a bridge between the Basic Software (BSW) layer and the Application layer. The purpose of the RTE layer is to provide communication services to the Application layer's software components by abstracting the complexity within the BSW layer.

The third and bottom layer is BSW, the largest layer with 214 associated documents. The BSW layer is divided into three sub-layers detailing services, ECU abstraction, and microcontroller abstraction over 18 sub-modules of various services, abstractions, and drivers. The service modules facilitate the interaction between the application layer and the BSW layer, such as diagnostics and communication. The abstraction modules cover both microcontroller and ECU abstraction, providing an interface for the hardware of ECUs and simplifying the accessibility of hardware. The driver modules allow the integration of non-standardized hardware drivers into AUTOSAR. All these modules are critical for automotive software and encompass a wide range of services, cornerstones, and abstractions [5].

### 3.3 The RAG Setup

This section describes the experimental setup. The first part begins with data pre-processing, followed by the embedding model and chunking strategy. The next part describes the utilized vector database, followed by the retrieval process and prompting methods. Lastly, the section presents the utilized LLMs and the evaluation method.

### 3.3.1 Data Pre-processing

The AUTOSAR collection and its associated specifications are in the format of PDF, incompatible with the RAG-framework. Therefore, the data was converted into a text-based format utilizing the Python libraries `PDFPlumper` and `Camelot` from `PyPI`. The libraries provided tools for automating extraction and conversion, facilitating the process. The pre-processing activity aimed to remove redundancies, ambiguity, and distractions to enhance the index structure and the data granularity. Existing research and literature emphasize the significance of pre-processing in improving retrieval performance and generation quality, presented in Section 2.6.2.1.

We defined an approach for extracting unstructured and structured data, processed using different libraries. `PDFPlumper` was observed to represent text well while tables more poorly, could not distinguish between the two categories. Conversely, `Camelot` was observed to more accurately extract tables, combining the libraries supported both data categories.

The approach used boundary boxes to avoid duplications and ensure that content was not extracted multiple times. Boundary boxes enabled automatic identification of specific regions to be isolated from being extracted once again. Unstructured data was processed utilizing `PDFPlumper`, while structured data was extracted using `Camelot`, which was transformed into `Pandas DataFrames` and then extracted as key-value pairs. This aimed to preserve contextual relevance in plain text format, the result of this approach is depicted in Figure 3.3.

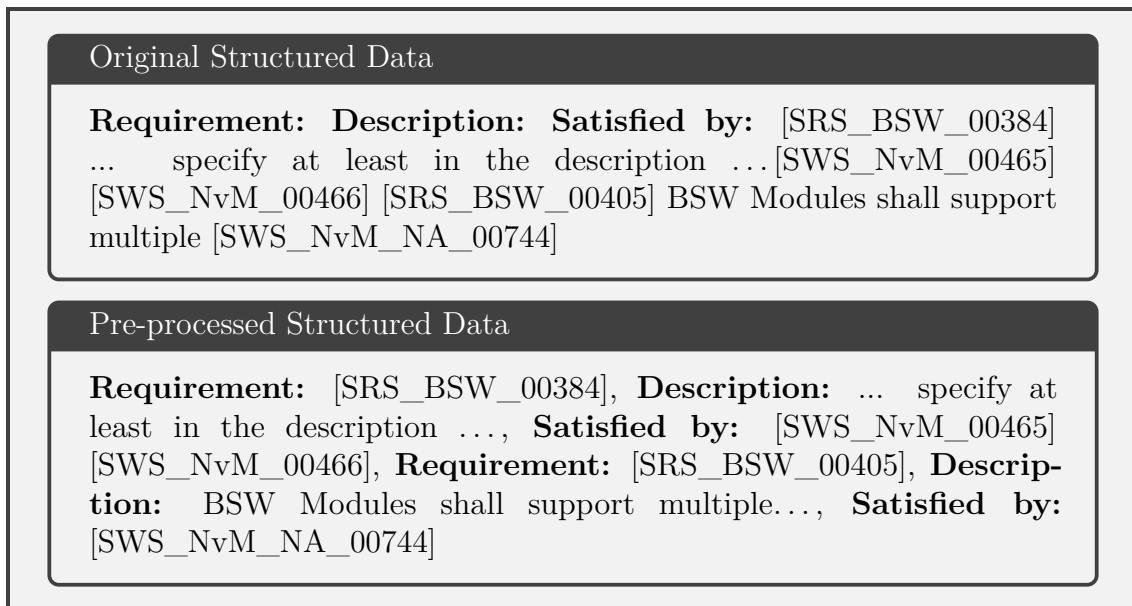


Figure 3.3: Structured data before and after pre-processing.

The top box in Figure 3.3 shows an example of structured data extracted without any processing, while the bottom box presents the same data extracted using our defined approach. This aimed to organize the data to clearly represent the relationships and

maintain relevance. This with the potential to enhance the relevance and utility for retrieval and generation.

This activity included removing footnotes, headers, figure texts, front pages, specification titles, document IDs, and release dates considered as noise and distraction for both unstructured and structured data. This was mostly done by pattern matching using the library `RE`.

### 3.3.2 Embedding Model and Chunks

The extracted, converted, and processed PDF files underwent additional processing, aimed to improve the capability of the RAG model. The obtained text-format files were chunked into smaller, more manageable text pieces. Focusing on smaller text pieces enables better relevant retrieval. For splitting and obtaining chunks, we used the automated tool `RecursiveCharacterTextSplitter` provided by the `LangChain` [87] library. This tool enables tuning of parameters such as chunk size and chunk overlap. We observed setting the chunk size to 1550 characters and chunk overlap to 150 characters yielded the best performance. There is no one-size-fits-all, and our tuning was solely based on the average count of characters per page and the GTE embedding model's maximum token limit of 512.

The GTE embedding model was used to transform the pre-processed chunks into numerical representations in the form of vectors. It is a common approach in NLP tasks, presented in Section 2.4. Moreover, this was an essential step in this experiment, as the embedded representation enabled the chunks of AUTOSAR to be efficiently retrieved based on semantic similarity. The GTE embedding model was chosen from the Massive Text Embedding Benchmark (MTEB) chart [88] and is proven to be good for several different embedding purposes.

### 3.3.3 Vector Database

The processed text of AUTOSAR, extracted metadata, and keywords from each chunk were stored in a vector database along with its embeddings. Weaviate [89] was chosen as the vector database in this experiment. Primarily for its robust features for handling high-dimensional data and support for both semantic searches and keyword-based searches. Weaviate additionally offers an open-source, cloud-based solution which is critical for scalability and accessibility.

Python was utilized to configure the database schemas and manage the created database objects. The `Weaviate Python` library and its specific tool `Client` were used for different data management tasks, such as adding, removing and modifying data objects. The library `NLTK` and `Collection` were used for counting keywords when loading and adding data objects into the Weaviate database.

### 3.3.4 Retrieval

The experiment included a systematic approach and strategy to finding the most effective retrieval technique. During the experiment, the implementation of retrieval

was documented, a crucial component for efficiently exploring the vector database after relevant documents.

### 3.3.4.1 Selection of Retrieval Technique

Research and literature suggest advantages in implementing a keyword-based search, ideally formulating a hybrid retrieval strategy as presented in Section 2.6.2.2. Potentially in cases when dealing with specialized and technical terms, common within AUTOSAR. This combination aims to synergize the strengths of both methods, potentially enhancing the overall retrieval performance. Consequently, we developed a benchmarking strategy to ascertain the most effective retrieval technique for our framework.

The benchmark involved randomly extracting chunks from the vector database, followed by prompting GPT-4 to generate a question that explores the specific content of these chunks. The choice of utilizing GPT-4 was predicated on its capacity to automate the question-generation process as a substitute for having subject matter experts manually craft questions. The relevance of these generated questions was validated by a domain expert. An illustrative example of the query generation template is provided in Appendix A.

The top 15 most similar documents were retrieved from the vector database, across four distinct search techniques: vector-based search, BM25 search, hybrid search 1, and hybrid search 2. Hybrid search 1 was implemented with relative score fusion, which combines the scores from the different techniques. The alpha parameter, indicating the weight attributed to vector similarity in relation to key-word, was set at 0.7. A decision we took made as we wanted to emphasis more on the semantic meaning than key-word relevance. Hybrid search 2 had the alpha parameter set to 0.5, reflecting a more balanced weighting between vector similarity and keyword relevance.

Table 3.2: Comparison of the accuracy for different search algorithms.

Method	Top 1	Top 5	Top 10	Top 15
<b>BM25</b>	0.45 %	0.55%	0.70%	0.75%
<b>Vector</b>	0.05 %	0.05%	0.05 %	0.05%
<b>Hybrid Search 1</b>	0.05 %	0.40%	0.40%	0.45%
<b>Hybrid Search 2</b>	0.25 %	0.40%	0.50%	0.65%

The results shown in Table 3.2 provide a comparative analysis of accuracy among the search algorithms. This by focusing on their ability to retrieve the reference chunk among the top 1, top 5, top 10, and top 15 chunks. BM25 demonstrated the highest performance across all thresholds. It showed a gradual increase in accuracy from 0.45% at the top 1 to 0.75% at the top 15. This suggests that BM25 is consistently better at identifying relevant documents as the number of retrieved documents increased. The vector search has significantly lower accuracy, at only 0.05% across all thresholds. Hybrid search 1 and 2 show varying improvements over the pure vector search but do not surpass the BM25 method. Hybrid Search 2 performed better than

Hybrid Search 1, an indication of the role put on the keyword relevance. The decision to select Hybrid Search 2 as the primary search algorithm over BM25 is informed by scenarios in which users pose queries without specific keywords pertaining to the specification, focusing instead on implementation details. In such cases, it is essential that semantic meaning be weighted in the retrieval process for relevant documents. This approach ensures that the system accounts for broader contextual relevance, rather than relying solely on keyword matching.

### 3.3.4.2 Implementation of Retrieval

The retrieval process comprises embedding the query, searching through a two-class index in the vector database, and retrieving chunks. The GTE embedding model was employed to embed the query, ensuring consistency with the vector representations in the database. Subsequently, we delineated the vector space into two index classes, the RTE specification and BSW specifications. The rationale behind this structured separation is twofold. Firstly, it mirrors the layered software architecture of AUTOSAR, where different layers serve distinct purposes yet interact closely. Secondly, this approach mimics a graph structure as illustrated in Figure 3.4. This enables us to leverage the inherent connectivity within the data, a technique supported by the recent literature and research described in Section 2.6.2.2.

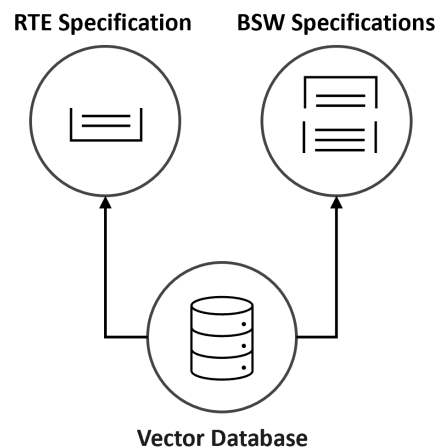


Figure 3.4: Multi-index in the vector database.

The number of chunks retrieved was determined by the parameter  $k$ , and in this experiment, we opted for setting  $k=15$ . This decision was guided by the theory and findings in Section 2.6.2.3, which highlighted that excessive retrieval could negatively impact the quality of the output. However, researchers are in disagreement, some mention only providing a selected set of documents such as  $k=10$ , while others suggest that more documents result in improved outputs. It is crucial to recognize that the ideal number of chunks is dependent on the specific context of the task. For our purposes, we selected  $k=15$  due to the complexity of the AUTOSAR specifications, which may require a broader context for thorough understanding.

This while recommending an optimal chunk size of 10. It is crucial to recognize that the ideal number of chunks is dependent on the specific context of the task. For our

purposes, we selected  $k=15$  due to the complexity of the AUTOSAR specifications, which may require a broader context for thorough understanding.

#### 3.3.5 Prompt Template

The design of the prompt template followed the literature and guidelines presented in Section 2.7 and was crafted to achieve three primary objectives. The initial step was to prime the LLM to act as an AUTOSAR assistant. This step ensured that the model’s responses were consistently aligned with the domain-specific knowledge, crucial for maintaining relevance. We instructed the LLM to always base its responses on the retrieved context, to ensure that the generated responses were relevant to the query and in pertinent information.

Additionally, the template covered instructions for the LLM to generate and include XML configuration examples, whenever applicable or requested. This ensured that any produced configuration strictly adhered to AUTOSAR related software applications, with the potential of enhancing the utility of the responses.

Furthermore, the template mandates the model to explicitly communicate the limitations of its output. This was in scenarios where the model encountered queries beyond its understanding or when the retrieved information was insufficient for an informed response. The goal of this instruction was to mitigate the risk of providing inaccurate or speculative information, aiming to enhance the reliability of the RAG model. The prompting template, covering the feature mentioned above, is provided in Appendix B.

#### 3.3.6 Large Language Model

In this experiment, two models were leveraged. Yi-34B-Chat-AWQ (Yi) was primarily utilized at the beginning of this experiment, due to its efficiency, fast transformer-based inference, open source, and reasonable quality for initial experimentation. GPT-4 was later utilized when most of the RAG architecture was in place, for its superior performance.

The prompt and retrieved context were augmented as input for the LLMs. The processed chunks were of size 1700, while the number of retrieved chunks was set to 15, a total of 25500 characters. This number can be approximated to 6375 tokens (4 chars = 1 token on average) as input for GPT-4. Moreover, no other configuration was done to the GPT-4 except setting the temperature to 0.5, as we wanted the generation to be a combination of deterministic and not too diverse throughout the experiment.

### 3.4 Evaluation for The RAG Setup

The evaluation was a critical part of our experimental setup. Throughout the process, we crafted a set of five questions in collaboration with subject matter experts. The

questions are outlined in Table 3.3, which mirrors the types of queries that could be posed within AUTOSAR CP for developers.

Table 3.3: Developed questions for evaluating the implemented models.

ID	Question
1	What steps need to be taken to implement CAN-FD considering enhancement of data transmission rates and payload capacity?
2	How can the SecOC module integrate with PDU to enhance security? Provide me with implementation details and the steps to be taken!
3	How to configure and handle CAN FlexRay data message, both in transmission and reception of a single frame? Please guide me in the process.
4	How to configure and set up a diagnostic messages routine and process step-by-step?
5	How to implement E2E (End-to-End) protect and security for a signal group?

### 3.4.1 Human Evaluation - Survey

The evaluation process involved human evaluators to analyze the correctness and quality of generated responses by three different setups outlined in Table 3.4. A panel of three subject matter experts was tasked with evaluating three distinct responses to the questions listed in Table 3.3 above.

Table 3.4: Comparison of the three different models.

GPT-4	Naive RAG	Developed RAG
No modifications	Off the shelf data extraction libraries GTE embedding model Vector search - Top 15 documents GPT4	Applied Preprocessing Pre-Retrieval strategies GTE embedding model Hybrid Search 2 - Top 15 documents GPT4

The first setup utilizes GPT-4 with fixed parameters and without any modifications or fine-tuning. This model serves as a baseline, to compare against the RAG setups. The second setup was a naive implementation of the RAG-framework, characterized by minimal text pre-processing, embedded using the GTE model, and stored in Weaviate. For retrieval, 15 chunks were retrieved based on vector similarity, and responses were generated by integrating GPT-4 using a basic prompt template. The developed RAG model is in-depth outlined in Section 3.3, with an overview provided in Section 3.1. This model distinguishes itself from the naive implementation through advanced techniques and customizations, covering pre-retrieval, retrieval, and post-retrieval.

Furthermore, the human evaluation was conducted through a survey, outlined in Appendix C. The survey was designed to collect insights and feedback from subject

matter experts. Three participants were instructed to review each response based on the four evaluative questions presented in Table 3.5. The outputs were anonymized such that respondents were unaware of which output corresponded to which LLM or RAG-model. This survey approach standardized the evaluation process and ensured impartiality. This evaluation mirrors the methodological process found in the literature described in Section 2.9 and centers on the evaluative criteria found in Table 3.5.

Table 3.5: Evaluative criteria and used survey questions.

<b>Criteria</b>	<b>Survey Questions</b>
Correctness	Does the response correctly address the question? Does the response contain any irrelevant information or inaccuracies?
Quality	How logically coherent do you find the reasoning in the response? Which response do you find the most compelling, and which do you find the least compelling?

The first question in Table 3.5 was designed to evaluate whether correct information was present in the responses. Complementary, the second question aimed to identify any factual inaccuracies or irrelevant information that potentially undermined the correctness of the responses. The logical progression and the flow of each response were included and evaluated in the survey, a response can be technically correct but lack clarity or effectiveness, which is crucial for its practicality. Additionally, we were interested in the preferences of the subject matter experts. The fourth question explored the compellingness by ranking the three responses from most to least for each question.

#### 3.4.2 Automatic Evaluation - RAGAS

The evaluation process incorporated the use of the automatic evaluation tool RAGAS. Useful in our experiment, where a traditional way of ground truth setup is unavailable and impractical to construct. The employed metrics include FA, AR, and CR.

### 3.5 Context Integration Experiment

This part of the study primarily investigates the consistency of data integration by LLMs.

Originally, our research aimed to explore how LLMs integrate information within the AUTOSAR framework. However, due to the technical complexity and the specialized knowledge required in the AUTOSAR domain, we narrowed our focus to evaluating the correctness and quality of responses using subject matter experts. Consequently, a revised experimental strategy was developed to employ a custom and simplified set of data. This adjustment allowed a broader investigation into the LLMs data

integration capabilities without the constraints of domain-specific complexities and knowledge.

### 3.5.1 Experimental Procedure

The LLMs are presented with a query along with two distinct documents, labeled as Document 1 (D1) and Document 2 (D2). These documents provide the necessary context for the queries, and each scenario is designed to test different aspects of the LLMs ability to process and integrate information. These scenarios are:

1. **Full Integration (FI)**: Queries in this category require the LLMs to synthesize and integrate information from both D1 and D2. This is expressed by the following condition in Equation 3.1:

$$FI(q) = \begin{cases} \text{True} & \text{if synthesis from both } D1 \text{ and } D2 \text{ is required for query } q \\ \text{False} & \text{otherwise} \end{cases} \quad (3.1)$$

2. **Single Document Synthesis (SDS)**: Queries in this category require the LLMs to integrate information from a single document when provided with both D1 and D2. The condition for SDS is defined as Equation 3.2:

$$SDS(q) = \begin{cases} \text{True} & \text{if the answer to } q \text{ relies solely on } D1 \\ \text{False} & \text{otherwise} \end{cases} \quad (3.2)$$

### 3.5.2 Custom Dataset

The customized and simplified dataset was specifically developed to circumvent the pre-encoded parametric knowledge of LLMs. Our goal was to ensure that the LLMs respond solely based on the context provided by us. Thus, the queries and the accompanying documents were crafted to be new to the model. Despite the inherent challenges due to the vast amount of encapsulated knowledge in LLMs, our experimental design assumes that both queries and documents are completely novel to the model.

The dataset is segmented into four categories. Each category contains 20 queries and each query is associated with two documents noted as D1 and D2.

1. **Combination Text (CT)**: This category evaluates the LLM’s ability to synthesize and integrate textual information from multiple sources. Queries in this category are designed to test how well the model can combine textual data to provide coherent and relevant responses.
2. **Combination Numerical (CN)**: This category assesses the LLM’s proficiency in integrating and interpreting numerical data presented within the documents. The focus is on how effectively the model can process and combine numerical information to generate accurate responses.

3. **Contradiction Text (CXT)**: This category tests the LLM’s skill in identifying and resolving textual contradictions within the provided documents. Queries here are crafted to include conflicting textual information, challenging the model to discern and reconcile these discrepancies.
4. **Contradiction Numerical (CXN)**: This category examines the LLM’s capability to handle and resolve contradictions in numerical data. The queries are designed to present conflicting numerical information, requiring the model to identify and address these inconsistencies accurately.

The goal of the "Combination" subsets is to evaluate the LLMs capacity to synthesize and integrate information from both text and numerical data presented in the documents. The ability to effectively synthesize and integrate information from diverse data types is critical for many practical applications. In real-world scenarios, data is often presented in multiple formats, and the ability of LLMs to handle this complexity directly impacts their usefulness.

Conversely, the "Contradiction" subsets are designed to evaluate the LLM’s proficiency in contradictions within the content. This aspect is important because real-world data is often inconsistent and ambiguous. The ability of LLMs to resolve contradictions and clarify ambiguities is crucial for their effectiveness in delivering accurate and reliable information.

Table 3.6 shows an excerpt of the manually created dataset used in our experiment. All documents were constructed to maintain a similar distribution, fostering an environment to analyze closely related sources. The first two examples shows queries that need to integrate information from both documents, in text or numerical format. In contrast, the third and fourth examples present contradictory information, where only Document 1 contains the correct answer, also in text or numerical format. The complete dataset is found in the link provided in Appendix D.

	<b>Query</b>	<b>Document 1</b>	<b>Document 2</b>
CT	How did the main characters’ parents feel in the movie Prustic 123?	The parents were sad when the main character died.	The parents were happy when the main character was revived.
CN	How many times did the main character punch someone in the movie Prustic 123?	The main character in the movie "Prustic 123" punched the bad guy 4 times.	The main character in the movie "Prustic 123" punched his best friend 2 times.
CXT	Where is Tracadonia located at?	Tracadonia is a city located next to Mapedonia and Capedonia.	Tracadonia has the highest mountain with an amazing scenery to view.
CXN	How many times did the main character punch someone in the movie Prustic 123?	The main character in the movie "Prustic 123" punched the bad guy 4 times.	The main character in the movie "Prustic 123" ate donuts 3 times.

Table 3.6: Example questions and documents from the dataset for the context integration experiment.

### **3.5.3 Integration Evaluation**

We evaluated the performance by calculating the accuracy for each specific category in the dataset. The ground truth, predefined for each query, served as the benchmark for comparison against the model outputs, which were analyzed for evidence of integration from one or both documents.



# 4

## Results

This chapter is divided into three sections. The first section details the computed RAGAS results for both the naive and developed RAG models, which presents the obtained scores of the two models. The second section presents the findings from the conducted survey, outlining identified insights. Lastly, the third section covers the obtained results from the context integration experiment.

### 4.1 Automatic Evaluation

The achieved scores for the naive and developed RAG models are summarized in Table 4.1, comparing the performance of both models. The metrics were computed using the same set of five questions enlisted in Table 3.3, evaluated over five iterations for each question and model.

Table 4.1: Average RAGAS scores for the naive and developed RAG models.

Questions	Naive Model			Developed Model		
	FA	AR	CR	FA	AR	CR
E2E Protect & Security	0.707	0.708	0.041	0.733	0.722	0.142
Diagnostic Messages Setup	0.560	0.701	0.003	0.194	0.695	0.170
CAN Flexray Data Message	0.062	0.715	0.172	0.725	0.761	0.177
SecOC Module Integration	1.000	0.780	0.194	0.812	0.788	0.218
Implementing CAN-FD	0.504	0.792	0.013	0.789	0.766	0.111
<b>Average</b>	<b>0.567</b>	<b>0.739</b>	<b>0.085</b>	<b>0.651</b>	<b>0.746</b>	<b>0.164</b>

The naive RAG model achieved average scores of 0.567, 0.739, and 0.085 in FA, AR, and CR, respectively. CAN Flexray Data Message recorded the lowest FA score at 0.062, indicating minimal alignment with the provided contexts. Conversely, SecOC Module Integration achieved a score of 1 across all five iterations, suggesting complete alignment with the contexts. E2E Protect & Security, Diagnostic Messages Setup, and, Implementing CAN-FD achieved 0.041, 0.003, and 0.013, respectively in CR. These results imply that the naive model struggles to ensure alignment between the contexts and queries. More specifically, CAN Flexray Data Message scored 0.062 in FA, 0.715 in AR, and 0.172 in CR, indicating that while the generated answers are generally related to the question, they do not accurately reflect the provided context and exhibit poor contextual relevance.

The developed RAG model demonstrates slight improvements with average scores of 0.651 for FA, 0.746 for AR, and 0.164 for CR. There is a significant improvement in the CR score identified for Diagnostic Messages Setup, which increased to 0.170, while accompanied by a decrease of FA to 0.194. This is in comparison to the scores of the naive RAG model at 0.003 and 0.560, respectively. The highest scores are achieved for SecOC Module Integration, 0.812 in FA, 0.788 in AR, and 0.218 in CR. The lowest FA score is found for Diagnostic Messages Setup at 0.194, and the lowest CR score is obtained for Implementing CAN-FD. This is while the achieved AR scores remain stable for the developed RAG model.

#### 4.1.1 Faithfulness

The achieved scores of the RAGAS metric FA for the naive and the developed RAG models are illustrated in Figure 4.1 and in Figure 4.2. The figures present the distribution of obtained scores over five iterations using box plots.

The FA scores of the naive RAG model are presented in 4.1, indicating a significant variability across the questions. Specifically, there are outliers in E2E Protect & Security and CAN Flexray Data Message, while Diagnostic Messages Setup and Implementing CAN-FD exhibit a wide range of scores, from low to high. This variability and the outliers suggest that the RAGAS framework struggles to consistently evaluate whether the generated responses correctly reflect and align with the contexts.

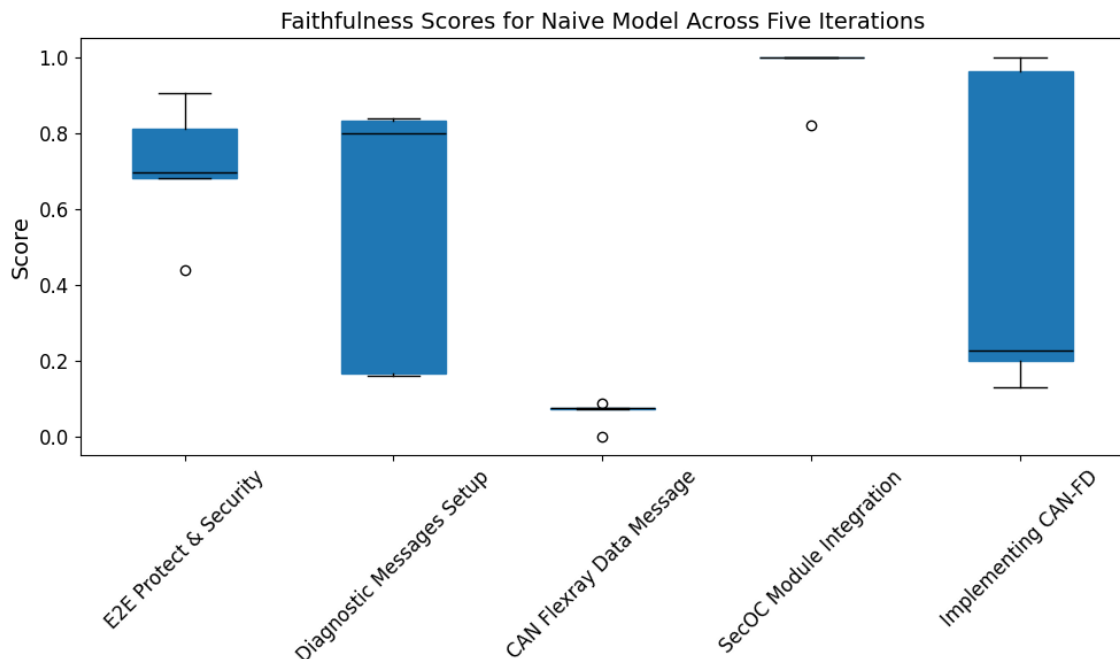


Figure 4.1: Box plot presenting the distribution of FA scores for the naive RAG model.

Figure 4.2 illustrates the distribution of FA scores for the developed RAG model. Overall, the box plot shows a reduction of variability across the question. For

instance, Diagnostic Messages Setup had a wider range of scores for the naive RAG model. Similarly, Implementing CAN-FD, additionally had a low average and is now exhibiting more consistent scores and reduced variability.

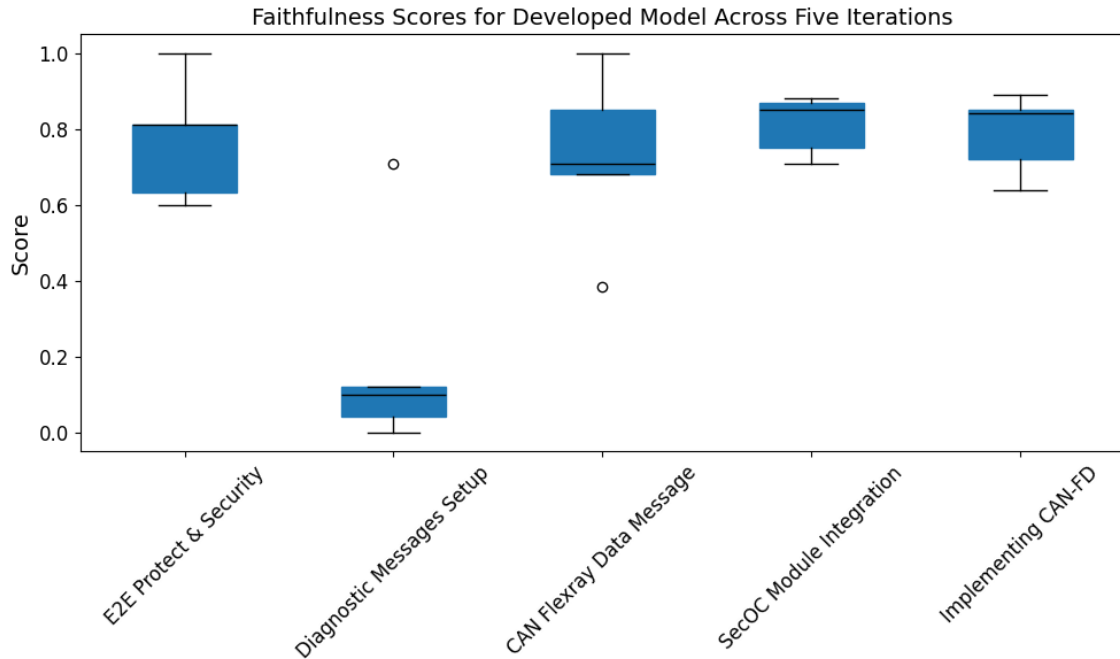


Figure 4.2: Box plot presenting the distribution of FA scores for the developed RAG model.

### 4.1.2 Answer Relevance

The achieved AR scores obtained by the RAGAS evaluation for both the naive and the developed RAG models are illustrated in Figure 4.3 and Figure 4.4. The naive RAG model demonstrates slight variability, notable outliers are present in SecOC Module Integration, which also are found for Diagnostic Messages Setup and Implementing CAN-FD.

Furthermore, despite the variations illustrated in Figure 4.3, the achieved AR scores for all five questions are high. SecOC Module Integration achieved the highest score indicating high answer relevance to the questions while Implementing CAN-FD obtained the highest score on average, suggesting consistent performance. Moreover, the lowest AR score is found for Diagnostic Messages Setup, which additionally achieved the lowest score on average.

The AR scores for the developed RAG model are presented in Figure 4.4. An increase in variability can be identified in a comparison to the naive RAG model, for instance, CAN Flexray Data Message shows a wider range of AR scores, which also is the case for SecOC Module Integration and Implementing CAN-FD. However, there are also outliers present for the developed RAG model, such as for E2E Protect & Security, Diagnostic Messages Setup, and Implementing CAN-FD.

## 4. Results

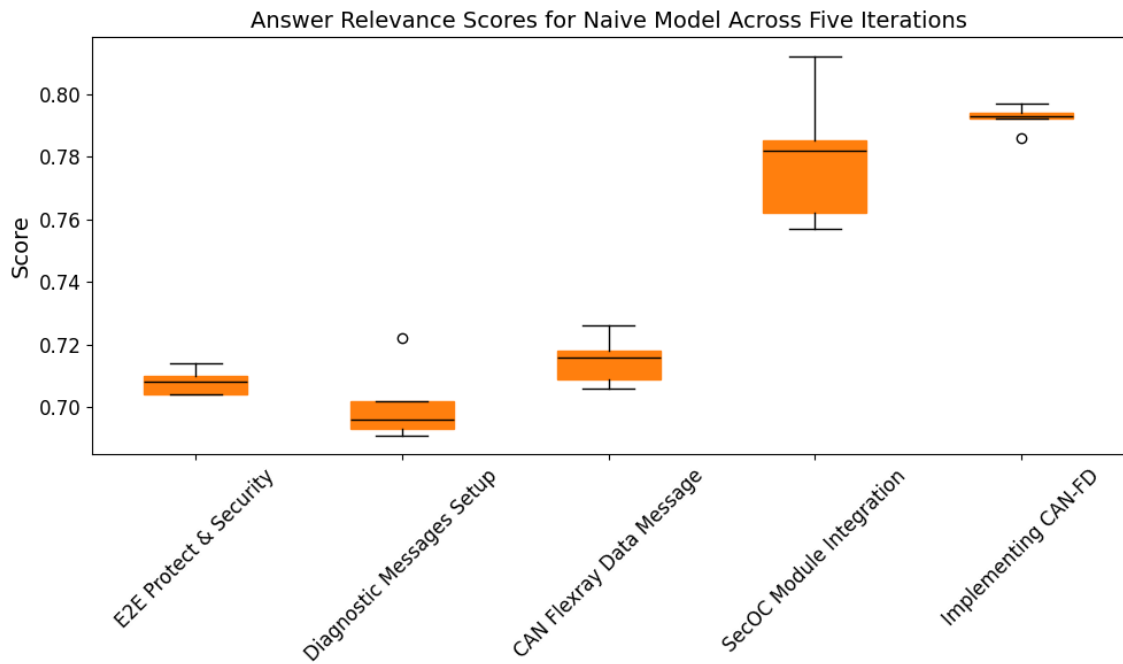


Figure 4.3: Box plot presenting the distribution of AR scores for the naive RAG model.

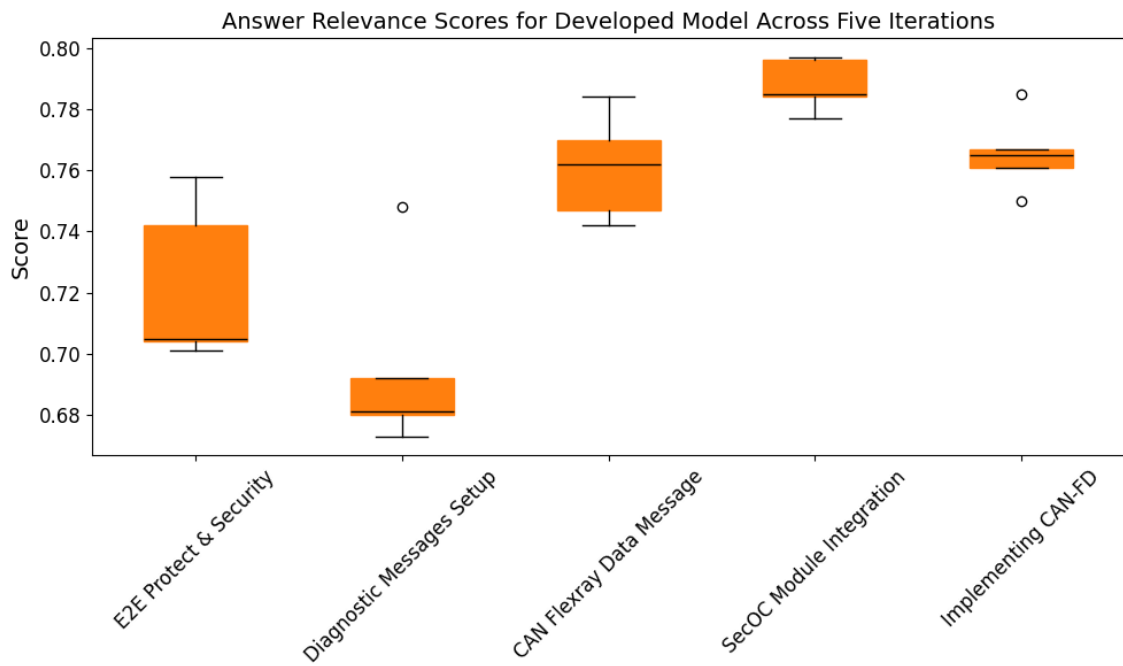


Figure 4.4: Box plot presenting the distribution of AR scores for the developed RAG model.

### 4.1.3 Context Relevance

In Figure 4.5 and Figure 4.6 the achieved scores in CR for the naive and the developed RAG model are presented, respectively. As shown in Figure 4.5, the scores of the naive RAG model vary, particularly notable for CAN Flexray Data Message and SecOC Module Integration. This indicates inconsistencies considering how the RAGAS framework evaluates the relevance of the retrieved contexts. However, Diagnostic Messages Setup shows minimal variation while achieving low CR scores for all five iterations. This is also identified for E2E Protect & Security and Implementing CAN-FD, both achieving low scores with less variation compared to CAN Flexray Data Message and SecOC Module Integration.

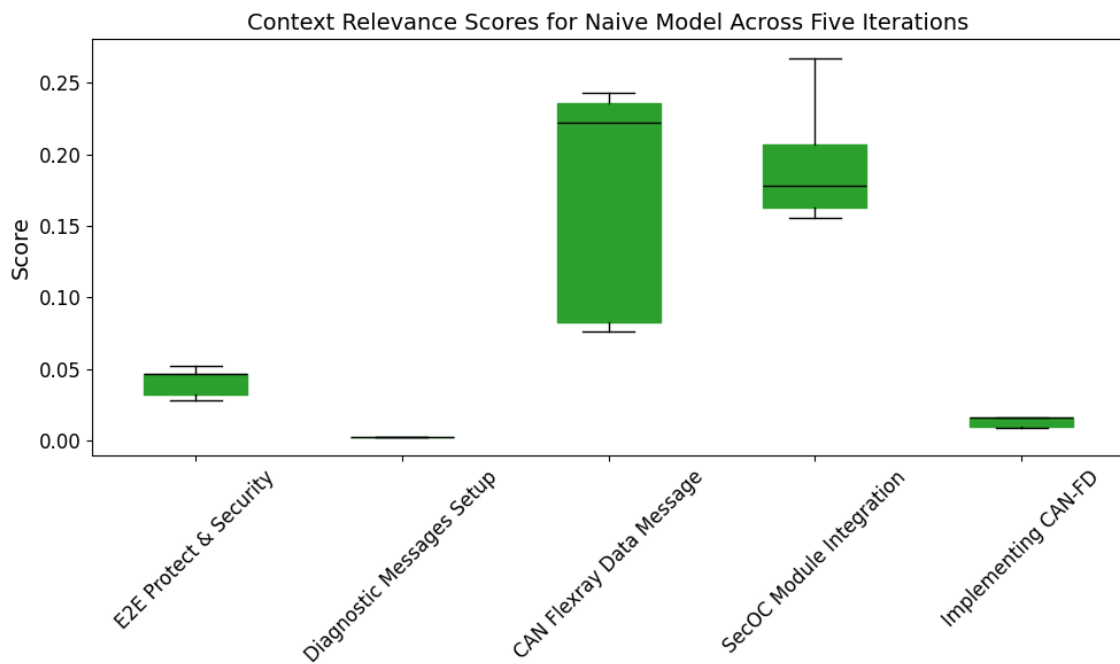


Figure 4.5: Box plot presenting the distribution of CR scores for the naive RAG model.

Furthermore, the box plot in Figure 4.6 shows the distribution of CR scores for the developed RAG model. The overall CR score is higher across all questions on average, while CAN Flexray Data Message and SecOC Module Integration show improvement considering the range of values in comparison to the naive RAG model.

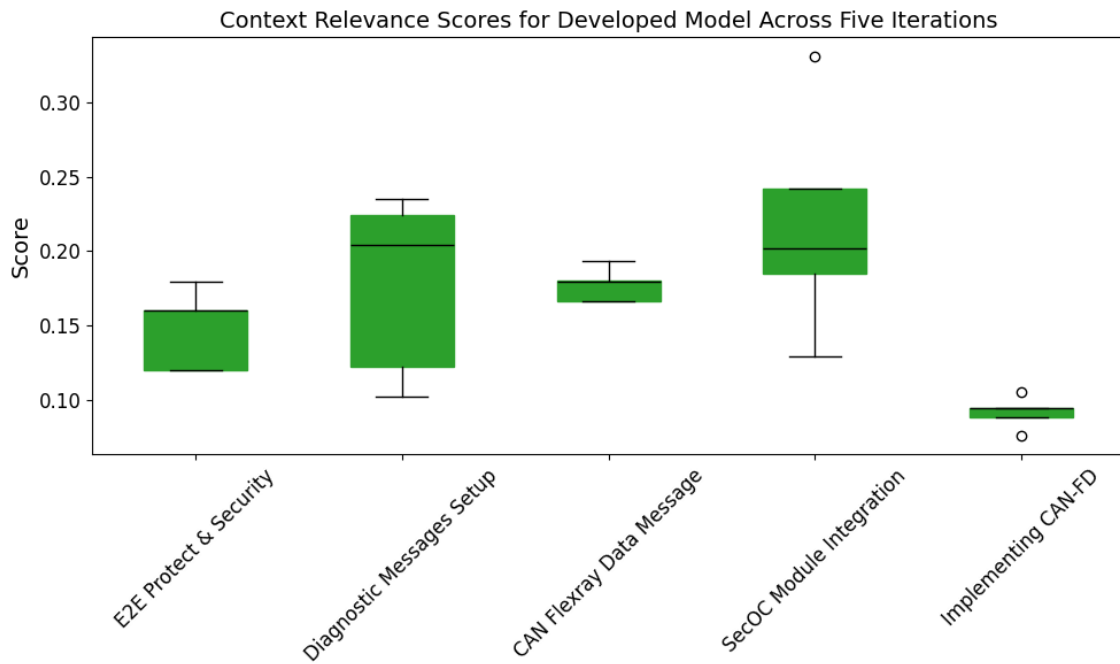


Figure 4.6: Box plot presenting the distribution of CR scores for the developed RAG model.

## 4.2 Human Evaluation

This section is structured according to the questions of the survey, the respondents' answers are presented and summarized in the subsections below.

### 4.2.1 Does the response correctly address the question?

The first respondent noted variations in the level of detail across the models. The developed RAG model was perceived as accurate with a lot of details, oriented towards a software developer using AUTOSAR. Conversely, the naive RAG model and GPT-4 were still perceived as accurate but of fewer details, resembling an architecture or upper management level to-do list. Moreover, the respondent mentioned that GPT-4 failed to answer the AUTOSAR queries related to SecOC module integration, diagnostic Messages Setup, and E2E protect & security. The responses either were too focused on introducing rather than providing implementation details or described something unrelated to AUTOSAR.

The second respondent evaluated that all responses partially addressed the questions. The questions were perceived to be skimmed through and the responses therefore gave general guidelines for the naive RAG model and GPT-4. This was also noted for the developed RAG model, where the respondent mentioned that it is rather shallow for implementation. The developed RAG model provides technical details for answering SecOC Module Integration, while the naive RAG model mentions key points but fails to detail the specifics. The GPT-4 addresses key points but no specifics on how to integrate them.

The third respondent mentioned that GPT-4 deviated from AUTOSAR considering two of the five queries and moderately addressed the questions. This while the naive model and developed model were both perceived to be similar and correctly address all the questions. However, GPT-4 lacked specificity and was only correct to some extent, while the developed RAG model was perceived to provide relevant responses for all five queries accurately. The naive RAG model was also relevant but less precise in comparison to the developed RAG model according to the third respondent.

In Figure 4.7, the specifics of the evaluation from the first and second respondents are highlighted. Presenting each model's ability to correctly address the question "Implementing CAN-FD".

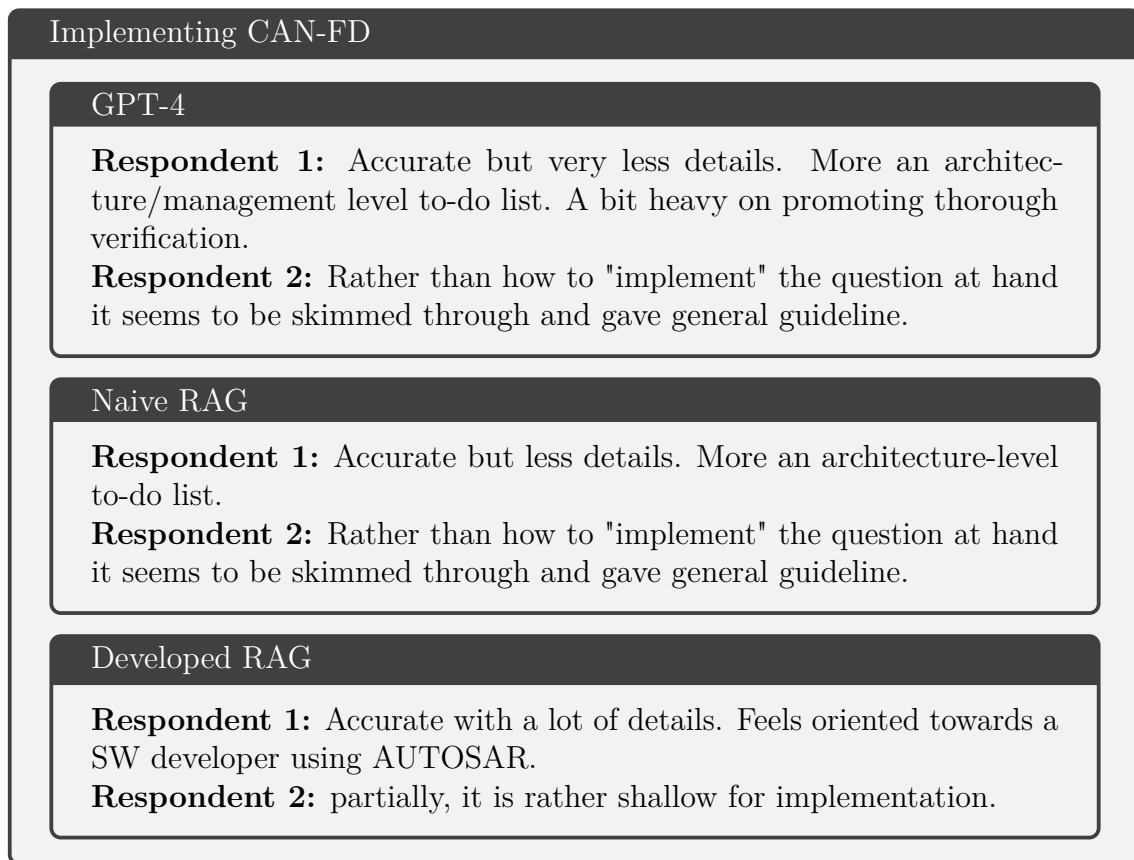


Figure 4.7: Survey responses on correctly addressing the question Implementing CAN-FD.

#### 4.2.2 How logically coherent do you find the reasoning in the response?

Notably, the first respondent answered that most of the responses from all the models were described in a logical order, seemed reasonable, and were of logical workflow. However, elaborated that the developed RAG model in one instance was a bit high and low, while GPT-4 was considered to have a logical flow on project plan level. The naive RAG model was in that particular instance described as having a logical

workflow.

The second respondent mentions that there are hundreds of configuration parameters and that the responses of the developed RAG model mention just a couple. This raised the question: are these parameters the most important or the only ones that are needed? However, the steps that are provided were perceived to be logical, but the parameters could confuse due to lack of clear detail. The naive RAG model and GPT-4 were noted to lack in terms of organization and workflow. For instance, the topology design is incorporated where it should already be established when asking how to implement it. The respondent gets the feeling that it seems to not fully grasp AUTOSAR development and as it is now it lacks good organization. The GPT-4 responses do not follow a clear progression could confuse beginners and is shallow for developers. Furthermore, the respondent mentions that implementation is about several parameters and is usually intuitive, module by module.

The third respondent noted that the responses of the developed RAG model are detailed and concise, logical and well-structured, including sequential steps making it easy to follow. The naive RAG model is perceived to also be detailed and cover the necessary steps effectively. However, the responses from GPT-4 were noted to be logical, but frequently a bit lengthy. The structure was clear but lacked some queries such as diagnostic messages setup and E2E Protect & Security.

The comments left by two of the subject matter experts are presented in Figure 4.8. Highlighting the differences in opinions among the first and the second respondents considering the reasoning for the question covering "Implementing CAN-FD" and the three models.

### **4.2.3 Does the response contain any irrelevant information or inaccuracies?**

The first respondent noted that the documentation and training for the naive RAG model and GPT-4 felt too obvious and not specific. However, the naive RAG model seemed to be relevant, but heavily promoted practices and "side activities" such as testing, deployment, and maintenance. This was considered irrelevant and excessive by the respondent. Notably, GPT-4 had responses that were perceived to be completely irrelevant and responses that were a bit redundant. The developed RAG model showed no major irrelevancies other than one of the responses had the wrong order of steps.

The developed RAG was considered to be satisfactory by the third respondent, merging bullet points could increase the relevance, while others not needed. It was understood to not be explicitly irrelevant but does not completely cover the complexity in AUTOSAR, for instance, considering diagnostics messages. Moreover, both GPT-4 and Naive RAG included irrelevancies and lacked the necessary workflow. Additionally, less precise explanations were done by the naive RAG model while GPT-4 deviated from AUTOSAR. The second respondent further elaborated that the responses provided by GPT-4 included unnecessary info and lacked considering details, some points were irrelevant and contained repetitive information. The naive

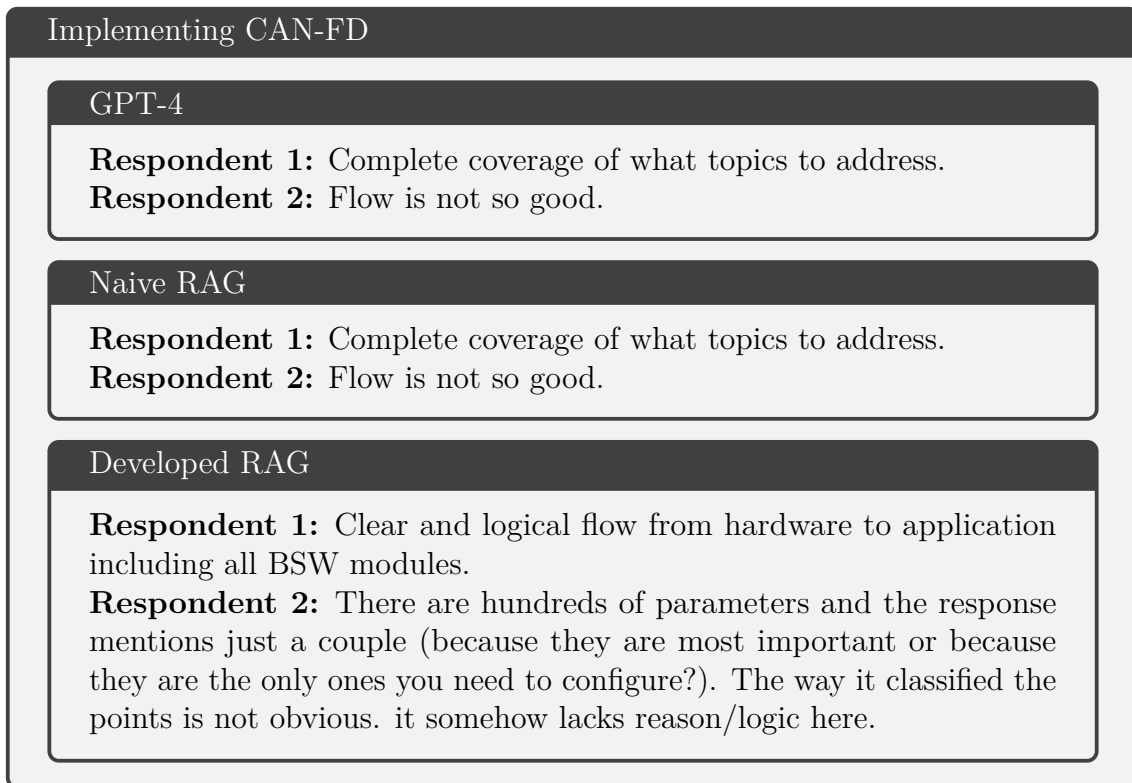


Figure 4.8: Survey responses on reasoning for the question Implementing CAN-FD.

RAG model also contained repetitive/redundant information and was perceived to be too broad.

Furthermore, the third respondent noted that the developed RAG model was noted to not consist of any inaccuracies, or minimal irrelevant details such as repeated information but were not considered as notable remarks. The naive model is also considered as to the point and consists of relevant information, but is perceived as a bit lengthy. Therefore, longer than necessary and is not that effective. GPT-4 was noted to include redundant elements and be lengthy, contributing to the lack of its effectiveness, and in two cases it was perceived to contain inaccuracies such as that the response and the AUTOSAR query are not aligned.

In Figure 4.9 comments from the first and the third respondents are presented for the question concerning "E2E Protect & Signal". The two respondents are aligned considering the three models, however, there are notable differences between GPT-4 and the developed RAG model.

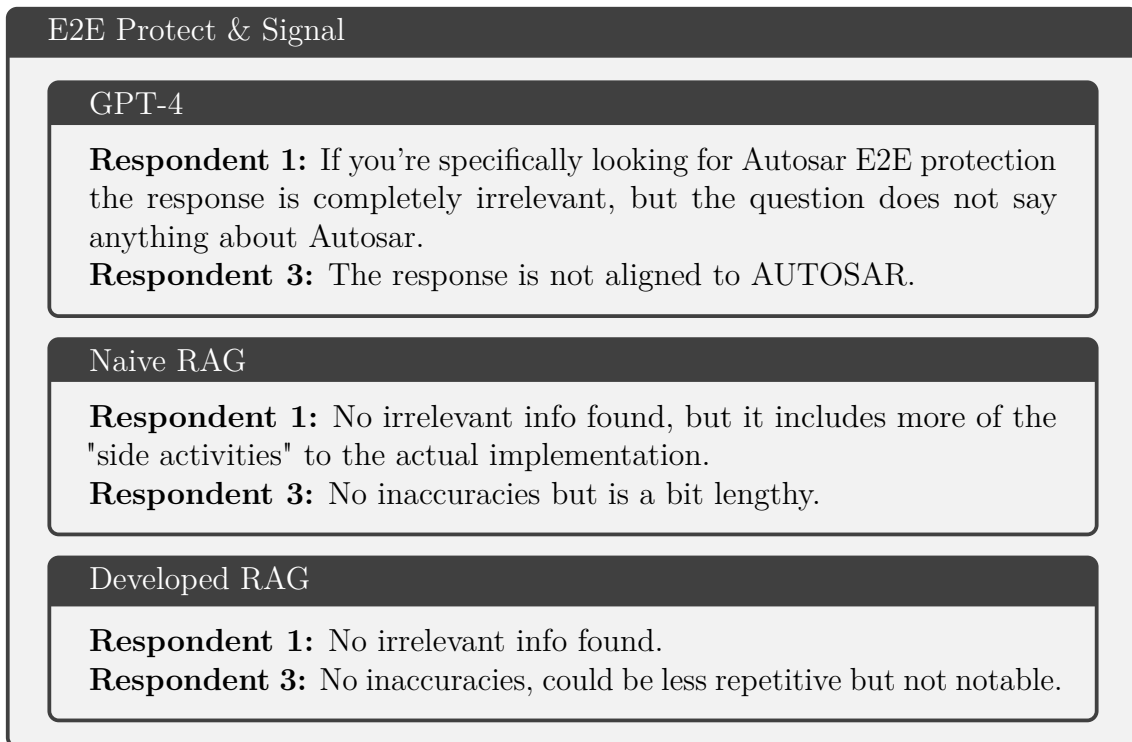


Figure 4.9: Survey responses on irrelevant information and inaccuracies for the question E2E Protect & Signal.

### 4.2.4 Which response do you find the most compelling, and which do you find the least compelling?

The first respondent stated that the developed RAG model would be preferred as a SW developer, naive RAG for an architect, and GPT-4 for managers. However, the respondent noted that the developed RAG model provided the best response for Diagnostic Messages Setup and Implementing CAN-FD. The naive RAG was chosen for E2E Protect & Security as it included info about the E2E profile. Additionally, the naive RAG model was selected twice as the most compelling by the first respondent. It was very even for CAN Flexray Data Message and mentioned that GPT-4 included parts of configuration tools and formats which was considered as good.

The second respondent mentioned that the developed RAG model is the best among the three, but does not fully meet the expectations for a technical implementation guide. The respondent elaborated further that the developed RAG model was closer to details than the others but still fails to provide implementable details. An AUTOSAR developer would expect more technical details and depth, for instance, by discussing the configuration parameters and knowing the workflow well according to the second respondent.

The third respondent was consistent considering the most compelling response among the five AUTOSAR queries. Without leaving a note on why, it was noted that the developed RAG model had the most compelling responses to the queries.

Figure 4.10 illustrates a pie chart showing the distribution of the most compelling model. GPT-4 was never considered the most compelling model for any response and question among the respondents. However, the naive RAG model was selected twice while the developed RAG model provided the most compelling answer 14 times. This sums up to 14 answers considering the most compelling model, due to that the responses for "SecOC Module Integration" were too even according to the first respondent.

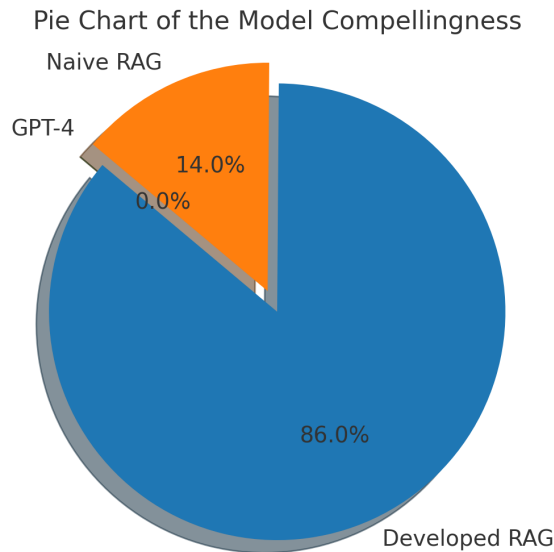


Figure 4.10: Pie chart presenting the compellingness of the responses.

### 4.3 Context Integration

The performance of the three models: Yi-34B-Chat-AWQ (Yi), GPT-3.5-turbo, and GPT-4 was evaluated across four distinct datasets, with each dataset targeting different aspects of text and numerical data combinations and contradictions. The results of this evaluation are summarized in 4.2.

Table 4.2: Obtained results from context integration.

Dataset	Yi	GPT-3.5-turbo	GPT4
CT	0.9	0.45	1
CN	0.95	0.75	0.85
CXT	0.9	0.9	0.95
CXN	0.95	0.9	0.95

In the CT dataset, Yi achieved an accuracy of 0.90, indicating a high ability to identify all relevant instances. However, GPT-3.5-turbo showed significantly lower

## 4. Results

---

performance, with an accuracy of only 0.45, suggesting difficulties in handling this specific type of data. In contrast, GPT4 performed exceptionally well, thereby identifying all positive instances without missing any.

For the CN dataset, Yi still showed strong performance with an accuracy of 0.95. GPT-3.5-turbo improved significantly compared to its performance in the text dataset, reaching an accuracy of 0.75. GPT-4, while generally robust, scored slightly lower in this dataset with an accuracy of 0.85, indicating some challenges in handling purely numerical data in comparison to textual data.

In the CXT dataset, both Yi and GPT-3.5-turbo displayed high levels of accuracy at 0.90, demonstrating their effectiveness in managing text data with contradictory elements. GPT-4 slightly outperformed the other models with an accuracy of 0.95, underscoring its advanced capability to discern and correctly classify more subtle textual nuances.

The CN dataset saw very high recall scores for all models, with both Yi and GPT-4 achieving an accuracy of 0.95. GPT-3.5-turbo, while slightly lower, still performed commendably with an accuracy of 0.90.

# 5

## Discussion

This chapter is structured to analyze the performance of the three settings of LLMs within the AUTOSAR framework. The insights gained from the conducted survey are discussed, followed by the achieved results from the RAGAS framework. Thereafter, the findings from the context integration experiment are discussed. Lastly, the chapter highlights ethical considerations and discuss threats to validity

### 5.1 From Generalist To Specialist

The results from the survey, which compares the output from the developed RAG model, the naive RAG model, and GPT-4, provide crucial insights into their functional dynamics and relative performance in handling complex technical queries within the AUTOSAR framework. The respondent's feedback emphasizes key themes essential for developing and optimizing LLMs and RAG models for specialized technical applications.

Across the responses, a consistent observation is the variation in the level of detail and technical accuracy provided by the models. The developed RAG model is more often recognized for its accuracy and detail, particularly in responses that require deep technical knowledge. This model's effectiveness appears to be due to its likely access to a specialized and rich knowledge base, combined with a retrieval mechanism to the specificities of AUTOSAR. Preliminary evaluations support this hypothesis, though further testing is required to confirm these findings.

Conversely, while accurate, the naive RAG model and GPT-4 offer less granularity in their responses. They instead provide information akin to general architectural overviews or managerial to-do lists. Reflects a broader but less specialized knowledge base, which, while beneficial for a range of topics, struggles with the depth required for specific technical implementations in software engineering.

Additional themes from the survey suggest a nuanced perspective on the specificity of the implementation provided by the models. Respondents noted that the developed RAG model, while detailed, sometimes lacks actionable guidance, suggesting a disparity between detailed knowledge and its application in practical steps. This feedback aligns with the second respondent's concern about the relevance and importance of the configuration parameters mentioned by the developed RAG model. There is uncertainty about how the model determines which parameters to emphasize,

considering that, in reality, many different parameters must be factored in.

Furthermore, the logical workflow and organization of responses were pointed out as strengths for the developed and naive RAG models, indicating their potential utility in structured problem-solving scenarios. GPT-4's responses were occasionally marked by a lack of clear progression and redundancy, which could lead to confusion, particularly for newcomers to AUTOSAR.

Maintaining relevance to the query while avoiding unnecessary information is critical in technical settings. Respondents observed that GPT-4 and the naive RAG model sometimes included irrelevant or redundant information. It not only dilutes the effectiveness of the responses but also brings into focus the challenge of tuning LLMs to disregard irrelevant data and focus on the essentials of the query.

The preference for different models based on the user's role within the AUTOSAR ecosystem is particularly telling. The developed RAG model was mentioned to be more in the style for software developers, whereas the naive RAG model and GPT-4 could be more preferred by architects and managers, respectively. This indicates a separation in model utility based on user expectations and needs, suggesting that no single model currently fulfills all roles effectively.

The feedback reiterates a common dilemma in AI development, balancing generalizability and specialization. While GPT-4 offers broad coverage, it lacks the specific depth required to address detailed AUTOSAR questions. In contrast, the developed RAG model demonstrates superior performance in these niche applications and might be closer to achieving the necessary expertise in this domain, though it requires further testing.

### 5.2 RAGAS Under the Microscope

The observation of high AR scores across both models suggests that the generated responses are largely pertinent to the queries. But at the same time, the distinction between AR and factual correctness should be noted. A relevant answer is not inherently accurate or truthful, merely aligned with the query's theme. This distinction draws attention to the need for a nuanced interpretation of the obtained "high scores" with the RAGAS metrics.

The obtained CR scores present an intriguing dichotomy, showing the pertinence of the information within the output and penalizes redundancies. Our results demonstrate a limitation in this metric arising from its dependency on the length and completeness of the retrieved context. For instance, brief and accurate answers derived from extensive contexts may receive unfairly low scores, reflecting a methodological flaw rather than a true lack of relevance. This issue suggests that the optimal functioning of the context relevance metric would benefit from a dynamic retrieval system with dynamic chunk size. Such a system would adaptively select only the most pertinent documents, a technically challenging yet potentially transformative advancement for improving relevance assessments. The advantages could be tremendous, but the complexity of implementing such a system has not been achieved and requires further

work and research to produce.

The divergence in FA scores between the naive and developed models is particularly revealing. The consistently low FA scores in the naive model indicate a disconnect between the retrieved documents and their impact in generating relevant output. This could stem from inadequate retrieval algorithms or insufficient quality of the retrieved documents, which fails to enrich the generation process.

Conversely, the variability in FA scores for the developed model from 0 to 1 in some scenarios, suggests that while advanced techniques can enhance the integration of retrieved information, they do not guarantee consistent application. High FA scores in certain instances correlated with improved performance, as observed through human evaluation. Yet the presence of low scores alongside high ones within the same model points to potential inconsistencies in how retrieval integrations are executed.

The variability observed in the FA metric raises broader questions about the reliability of RAGAS metrics. The assumption that the language model used in the evaluation process possesses sufficient contextual understanding is a big premise, given the inherent stochastic nature of LLMs. This stochasticity means that while LLMs can produce highly faithful responses, their performance is unpredictable and non-deterministic, posing challenges to the consistency of RAGAS metrics.

This unpredictability was evident in our study, where the stochastic nature of the LLM led to variable results that did not consistently align with expected outcomes based on the input data. Such fluctuations undermine the integrity of RAGAS results, as the metrics may reflect random variability rather than the model's true capability to understand and process queries faithfully. Given these observations, we still believe that the RAGAS metric offers a valuable framework for assessing certain aspects of LLM performance, particularly in reference-free datasets. However, its application in complex tasks, such as those involved in AUTOSAR explainability, must be approached with caution as their reliability is heavily compromised.

### 5.3 Component Synergy Boosts RAG Performance

The implementation of a rigorous data-cleaning process likely contributed substantially to the reduction of contextual noise. This reduction is crucial as it potentially enhances the overall framework's performance by providing cleaner, more relevant data for subsequent processing stages. However, further validation is needed to confirm the extent of this noise reduction.

Our examination of various retrieval algorithms highlighted the superior performance of BM25. This algorithm, which emphasizes keyword-based searches, was particularly effective in our tests. It identified pertinent terms and keywords, which likely contributed to its outperformance of other evaluated algorithms. However, it is also important to recognize the limitations of our benchmarking test. The experiment was solely based on identifying a single reference document. In practice, AUTOSAR specifications are interconnected with multiple other specifications, often found in

various documents. The benchmarking test does not account for these connections. Therefore, capturing the semantic meaning and the connections between semantic meanings across all relevant documents is essential. Thus, the hybrid search algorithm represents a more promising avenue to be employed in our study as it yielded almost the same performance as BM25, and we believe it has the potential for enhancing the system’s adaptability and accuracy in handling multi-modal data.

Regarding retrieval precision, our findings reveal an interesting discourse on the trade-off between the breadth of document retrieval and accuracy. While precise retrieval can theoretically lead to optimal results, practical limitations such as computational resources and response times necessitate a balance. The concept of using a reranker becomes relevant here. The reranker allows for the initial retrieval of a broader document set, subsequently refined by having another LLM act as a judge to discern relevant and irrelevant documents. We briefly experimented with rerankers during the prototype phase and observed mixed results, which made us hesitant to incorporate them into the developed model. We still see the potential for enhancing the overall efficiency of the RAG pipeline by integrating a reranker for future research.

The development and refinement of prompt templates emerged as a vital component in the effectiveness of our RAG model. Prompt engineering, as evidenced in the literature [78]–[80], plays a pivotal role in the utility and performance of generative models. Our experimentation with prompt engineering yielded positive results, as indicated by the outputs. However, we found it challenging to design an optimal prompt template that accommodates a variety of user queries while maintaining the intent and clarity necessary for effective understanding and response generation. Thus, further research into streamlining optimal prompt template design is warranted.

### 5.4 Context Matters

The results from our context integration experiment provide a compelling insight into the performance of LLMs, specifically GPT-3.5-turbo, GPT-4, and Yi, across combination and contradiction datasets. This experiment brings forth the evolving capabilities of these models in handling complex information integration tasks and provides several critical areas for further exploration.

The observed performance disparity between GPT-3.5-turbo and its successors is noteworthy. GPT-3.5-turbo exhibited poor performance on the combination dataset but fared better on the contradiction dataset. In contrast, both GPT-4 and Yi demonstrated robustness across both datasets, with accuracy levels ranging from 0.85 to near perfection. This improvement in performance from GPT-3.5-turbo to GPT-4 and Yi can be attributed to enhancements in model architecture, training methodologies, and perhaps an increased training dataset size, which collectively contribute to their improved ability to synthesize and differentiate information from multiple documents.

The near-equal performance of GPT-4 and Yi on both datasets suggests significant improvements in their capabilities to integrate and distinguish relevant information derived from separate sources. An essential feature for tasks requiring nuanced

understanding and contextual awareness, such as legal and medical document analysis. Where the ability to combine or separate information from various documents accurately has a significant impact on the model's reliability.

The experimental setup highlighted the impact of context on the model's performance. While promising, current results also point to the potential benefits of testing these models with more complex and contextually rich datasets. Incorporating multiple documents that extend beyond single-sentence information to paragraphs may further challenge and refine the models' capacity for context integration. This could lead to significant strides in their explainability, a critical aspect often demanded by applications in fields where decisions must be transparent and justifiable.

Future research should consider several enhancements. Expanding the dataset complexity could provide deeper insights into the models' operational limits and capabilities. This involves increasing the number of documents and the diversity and depth of content within those documents to more closely mimic real-world scenarios where LLMs are expected to perform.

Secondly, the manual creation of the current dataset, while necessary for controlled experimentation, may benefit from automation and scaling. Employing automated processes to generate testing materials can help systematically explore a broader array of scenarios and reduce potential biases or limitations associated with manually curated datasets.

In conclusion, the experiment shows the significant progress made from GPT-3.5-turbo to GPT-4 and Yi in handling complex integration tasks within an RAG-framework. The findings emphasize the progress in model development and delineate the essential next steps for research in this domain. By pushing the boundaries of dataset complexity and contextual richness, we can better understand and enhance the explainability and utility of LLMs across various applications, paving the way for more sophisticated, reliable, and transparent AI systems.

## 5.5 Ethical considerations

The complex nature of LLMs can create difficulties in understanding how they generate specific outputs, which may result in users feeling less confident. Guaranteeing transparency in the decision-making mechanisms of these models is crucial for building trust. Thus, it is essential to create and incorporate tools that allow users to track and understand the outputs of the models.

Deploying LLMs in specialized domains such as AUTOSAR can have significant consequences. The importance of establishing clear accountability for the models' actions and decisions needs to be emphasized. Defining responsibility for the models' outputs and ensuring mechanisms are in place to address errors or misuse is critical for maintaining accountability. A robust framework should be created to outline the responsibilities of developers and end-users. The inclusion of feedback systems to identify and rectify issues is essential, with clear protocols to address any negative impacts resulting from the use of the models, ensuring accountability at every level.

## 5.6 Threats to Validity

In evaluating the performance of LLMs within the AUTOSAR domain, several threats to validity must be considered. This is to ensure the reliability and generalizability of the findings, as these threats span across internal and external validity.

### 5.6.1 Internal Validity

The subject matter experts familiarity with AUTOSAR might not reflect the average user's experience, potentially skewing the results. The participants becoming more familiar with the AUTOSAR framework over time could also skew their feedback on the models' performance, as their growing expertise might lead them to be more critical or lenient in their evaluations.

### 5.6.2 External Validity

An external validity threat for this thesis is the specificity of the AUTOSAR domain and the limited scope of the dataset used. The findings may not generalize to other classifications or applications outside automotive software development. This limitation is particularly pertinent given the unique technical requirements and standards within the AUTOSAR domain, which may not be present in other fields.

The controlled experimental settings may not fully capture the complexity and variability of real-world environments where these models are deployed. For example, the model's performance in a fixed setting with clean and well-defined queries might not translate to real-world scenarios where queries are more ambiguous, and data is noisier. The limited number of documents used in the context integration experiment also restricts the generalizability of the results.

# 6

## Conclusion

This research has examined the outputs of the developed RAG model, the naive RAG model, and GPT-4, within the challenging domain of AUTOSAR. The feedback and analysis reveal nuanced insights into their capabilities and limitations in handling technically complex queries, ultimately contributing to the broader understanding of LLMs and RAG models for specialized applications.

Our findings demonstrate that while the developed RAG model showcases high levels of accuracy and detail, particularly useful for deep technical engagement, it sometimes lacks in providing fully actionable guides. This shows a disparity between theoretical knowledge extraction and practical application, suggesting areas for future enhancement. Conversely, GPT-4 and the naive RAG model, though generally accurate, often fail to match the specificity required for intricate software engineering tasks, highlighting their suitability for broader, less detailed managerial or architectural roles.

The application of RAGAS metrics has provided valuable insights into the models' performance. However, the inherent variability in these metrics, influenced by the stochastic nature of LLMs, calls into question their reliability for consistent performance evaluation. This variability suggests the need for more robust, context-aware mechanisms in model development and assessment, which could enhance both the precision and applicability of these models.

Our context integration experiment with models such as GPT-3, GPT-4, and Yi demonstrates the promising ability of LLMs to effectively synthesize and differentiate information across datasets, indicating their potential for complex information processing tasks requiring high levels of contextual understanding. However, the current limitations related to dataset complexity and the manual nature of dataset creation highlight essential areas for future research. Enhancing dataset diversity and utilizing automated generation techniques could provide more definitive insights into the models' operational capabilities and limitations.

Overall, this study articulates the critical balance between model generalizability and specialization, illustrating the profound impact of advanced RAG models in handling specialized tasks within technical domains such as AUTOSAR.

## 6.1 Future work

Future work should include more extensive evaluations by subject-matter experts to gain better understanding of each model’s performance. Given the specialized nature of AUTOSAR, finding experts is particularly challenging. The limitations of this study, particularly the small number of participants and queries, necessitate a more thorough investigation. One potential approach is to engage with the AUTOSAR organization for their feedback on the outputs. However, it’s imperative to acknowledge that human evaluation is resource-intensive. As Celikyilmaz et al. [82] noted, automatic metrics can offer significant advantages if they are consistent and reliable.

We observed inconsistencies with the RAGAS metric in outcomes across five iterations under the same context. Therefore, future research should explore alternative automatic metrics that are more robust, especially for datasets lacking ground truth.

Our study focused on extracting semi-structured data, including text and tabular data, but AUTOSAR also contains valuable graphical information. Expanding data extraction to include multi-modal models could enhance retrieval-process and improve output quality, providing more relevant and detailed information. This approach could also yield better explainable results, increasing user trust in the model. Implementing fine-tuning for the LLM or an embedding model could further improve the representation of the AUTOSAR domain, resulting in more comprehensible outputs. However, this requires a sufficient and high-quality dataset, which is not readily available in a suitable format. Developing such a dataset would involve collaboration with subject-matter experts. An emerging alternative in industry and research is generating synthetic datasets using LLM prompts. While the reliability of this method for a complex domain like AUTOSAR is uncertain, it warrants exploration to evaluate its effectiveness.

Regarding the RAG-framework, there is still limited knowledge on information integration. As proposed in our third research question, it is crucial to understand the limitations of information integration within LLMs or RAG-integrated LLMs. Future work should extend this experiment by developing a more complex dataset and comparing more documents. Although our experiment was limited to two documents, the expanding context windows in LLMs suggest the need to test whether RAG can effectively integrate a comprehensive set of information when the context is increased.

# Bibliography

- [1] D.-K. Choi, J.-H. Jung, S.-J. Koh, J.-I. Kim, and J. Park, “In-vehicle infotainment management system in internet-of-things networks,” in *2019 International Conference on Information Networking (ICOIN)*, 2019, pp. 88–92. DOI: 10.1109/ICOIN.2019.8718192.
- [2] T. Werquin, M. Hubrechtsen, A. Thangarajan, F. Piessens, and J. T. Mühlberg, “Automated fuzzing of automotive control units,” in *2019 International Workshop on Secure Internet of Things (SIOT)*, 2019, pp. 1–8. DOI: 10.1109/SIOT48044.2019.9637090.
- [3] R. N. Charette. “This car runs on code.” Accessed: 2023-11-26. (2019), [Online]. Available: <https://spectrum.ieee.org/this-car-runs-on-code>.
- [4] Y. Elkharaz, S. Motahhir, and A. Elghzizal, “Comparison between autosar platforms with functional safety for automotive software architectures,” in *Smart Embedded Systems and Applications*, 1st. River Publishers, 2022, ch. 2, pp. 21–32.
- [5] “Autosar.” Accessed: 2023-11-20. (2023), [Online]. Available: <https://www.autosar.org/>.
- [6] M. Fusani and G. Lami, *On the efficacy of safety-related software standards*, 2014. arXiv: 1404.6805 [cs.SE].
- [7] S. M. Agren, R. Haldal, E. Knauss, and P. Pelliccione, “Agile beyond teams and feedback beyond software in automotive systems,” *IEEE Transactions on Engineering Management*, vol. 69, no. 6, pp. 3459–3475, Dec. 2022, ISSN: 1558-0040. DOI: 10.1109/tem.2022.3146139. [Online]. Available: <http://dx.doi.org/10.1109/TEM.2022.3146139>.
- [8] S. M. Ågren, E. Knauss, R. Haldal, and S. Martinez-Fernandez, “The impact of requirements on systems development speed: A multiple-case study in automotive,” *Requirements Engineering*, vol. 24, no. 3, pp. 315–340, Jul. 2019, ISSN: 1432-010X. DOI: 10.1007/s00766-019-00319-8. [Online]. Available: <http://dx.doi.org/10.1007/s00766-019-00319-8>.
- [9] S. Smith and M. A. Khalid, “Automated generation and integration of autosar rte configurations,” in *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2022, pp. 141–144. DOI: 10.1109/CCECE49351.2022.9918435.
- [10] S. Martínez-Fernandez, C. P. Ayala, X. Franch, and E. Y. Nakagawa, “A survey on the benefits and drawbacks of autosar,” in *2015 First International Workshop on Automotive Software Architecture (WASA)*, 2015, pp. 19–26. DOI: 10.1145/2752489.2752493.

- [11] G. Xiong, Q. Jin, Z. Lu, and A. Zhang, *Benchmarking retrieval-augmented generation for medicine*, 2024. arXiv: 2402.13178 [cs.CL].
- [12] B. Zhang, H. Yang, T. Zhou, M. Ali Babar, and X.-Y. Liu, “Enhancing financial sentiment analysis via retrieval augmented large language models,” ser. ICAIF ’23, , Brooklyn, NY, USA, Association for Computing Machinery, 2023, pp. 349–356, ISBN: 9798400702402. DOI: 10.1145/3604237.3626866. [Online]. Available: <https://doi.org/10.1145/3604237.3626866>.
- [13] Y. Gao, Y. Xiong, X. Gao, *et al.*, *Retrieval-augmented generation for large language models: A survey*, 2024. arXiv: 2312.10997 [cs.CL].
- [14] P. Zhao, H. Zhang, Q. Yu, *et al.*, *Retrieval-augmented generation for ai-generated content: A survey*, 2024. arXiv: 2402.19473 [cs.CV].
- [15] Y. Roh, G. Heo, and S. Whang, “A survey on data collection for machine learning: A big data - ai integration perspective,” *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, pp. 1–1, Oct. 2019. DOI: 10.1109/TKDE.2019.2946162.
- [16] T. Ghandi, H. Pourreza, and H. Mahyar, “Deep learning approaches on image captioning: A review,” *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–39, Oct. 2023, ISSN: 1557-7341. DOI: 10.1145/3617592. [Online]. Available: <http://dx.doi.org/10.1145/3617592>.
- [17] E. Cambria and B. White, “Jumping nlp curves: A review of natural language processing research [review article],” *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, 2014. DOI: 10.1109/MCI.2014.2307227.
- [18] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948. DOI: 10.1002/j.1538-7305.1948.tb01338.x.
- [19] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf).
- [20] D. S. Shah, H. A. Schwartz, and D. Hovy, “Predictive biases in natural language processing models: A conceptual framework and overview,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 5248–5264. DOI: 10.18653/v1/2020.acl-main.468. [Online]. Available: <https://aclanthology.org/2020.acl-main.468>.
- [21] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, “RealToxicityPrompts: Evaluating neural toxic degeneration in language models,” in *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 3356–3369. DOI: 10.18653/v1/2020.findings-emnlp.301. [Online]. Available: <https://aclanthology.org/2020.findings-emnlp.301>.
- [22] Y. Elazar, N. Kassner, S. Ravfogel, *et al.*, “Measuring and improving consistency in pretrained language models,” *Transactions of the Association for Computational Linguistics*, vol. 9, B. Roark and A. Nenkova, Eds., pp. 1012–1031, 2021.

- DOI: 10.1162/tacl\_a\_00410. [Online]. Available: <https://aclanthology.org/2021.tacl-1.60>.
- [23] Y. Liu, Y. Yao, J.-F. Ton, *et al.*, *Trustworthy llms: A survey and guideline for evaluating large language models' alignment*, 2024. arXiv: 2308.05374 [cs.AI].
- [24] Z. Ji, N. Lee, R. Frieske, *et al.*, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, Mar. 2023, ISSN: 1557-7341. DOI: 10.1145/3571730. [Online]. Available: <http://dx.doi.org/10.1145/3571730>.
- [25] T. Bickmore, H. Trinh, S. Ólafsson, *et al.*, "Patient and consumer safety when using conversational assistants for medical information: Observational study (preprint)," *Journal of Medical Internet Research*, vol. 20, Jul. 2018. DOI: 10.2196/11510.
- [26] X. Shen, Z. Chen, M. Backes, and Y. Zhang, *In chatgpt we trust? measuring and characterizing the reliability of chatgpt*, 2023. arXiv: 2304.08979 [cs.CR].
- [27] H. Touvron, T. Lavril, G. Izacard, *et al.*, *Llama: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 [cs.CL].
- [28] H. Touvron and L. Martin, *Llama 2: Open foundation and fine-tuned chat models*, 2023. arXiv: 2307.09288 [cs.CL].
- [29] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language models are few-shot learners*, 2020. arXiv: 2005.14165 [cs.CL].
- [30] OpenAI, : J. Achiam, *et al.*, *Gpt-4 technical report*, 2024. arXiv: 2303.08774 [cs.CL].
- [31] S. Ruciński, *Efficient language adaptive pre-training: Extending state-of-the-art large language models for polish*, 2024. arXiv: 2402.09759 [cs.CL].
- [32] C.-A. Li and H.-Y. Lee, *Examining forgetting in continual pre-training of aligned large language models*, 2024. arXiv: 2401.03129 [cs.CL].
- [33] Z. Gekhman, G. Yona, R. Aharoni, *et al.*, *Does fine-tuning llms on new knowledge encourage hallucinations?* 2024. arXiv: 2405.05904 [cs.CL].
- [34] M. Jeong, J. Sohn, M. Sung, and J. Kang, *Improving medical reasoning through retrieval and self-reflection with retrieval-augmented large language models*, 2024. arXiv: 2401.15269 [cs.CL].
- [35] K. Singhal, T. Tu, J. Gottweis, *et al.*, *Towards expert-level medical question answering with large language models*, 2023. arXiv: 2305.09617 [cs.CL].
- [36] J. Wei, X. Wang, D. Schuurmans, *et al.*, *Chain-of-thought prompting elicits reasoning in large language models*, 2023. arXiv: 2201.11903 [cs.CL].
- [37] A. Q. Jiang, A. Sablayrolles, A. Roux, *et al.*, *Mixtral of experts*, 2024. arXiv: 2401.04088 [cs.LG].
- [38] A. Louis, G. van Dijck, and G. Spanakis, "Interpretable long-form legal question answering with retrieval-augmented large language models," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 20, pp. 22 266–22 275, Mar. 2024. DOI: 10.1609/aaai.v38i20.30232. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/30232>.
- [39] S. Yue, W. Chen, S. Wang, *et al.*, *Disc-lawllm: Fine-tuning large language models for intelligent legal services*, 2023. arXiv: 2309.11325 [cs.CL].

- [40] N. Wiratunga, R. Abeyratne, L. Jayawardena, *et al.*, *Cbr-rag: Case-based reasoning for retrieval augmented generation in llms for legal question answering*, 2024. arXiv: 2404.04302 [cs.CL].
- [41] P. F. J. R. Foulds and S. Pan, *Ragged edges: The double-edged sword of retrieval-augmented chatbots*, 2024. arXiv: 2403.01193 [cs.CL].
- [42] W. Peng, G. Li, Y. Jiang, *et al.*, “Large language model based long-tail query rewriting in taobao search,” in *WWW ’24: The ACM Web Conference 2024*, May 2024, pp. 20–28. DOI: 10.1145/3589335.3648298.
- [43] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, *Self-rag: Learning to retrieve, generate, and critique through self-reflection*, 2023. arXiv: 2310.11511 [cs.CL].
- [44] A. J. Yepes, Y. You, J. Milczek, S. Laverde, and R. Li, *Financial report chunking for effective retrieval augmented generation*, 2024. arXiv: 2402.05131 [cs.CL].
- [45] D. E. Rumelhart and J. L. McClelland, “Learning internal representations by error propagation,” in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. 1987, pp. 318–362.
- [46] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, pp. 1735–80, Dec. 1997. DOI: 10.1162/neco.1997.9.8.1735.
- [47] A. Radford, J. W. Kim, C. Hallacy, *et al.*, “Learning transferable visual models from natural language supervision,” in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 8748–8763. [Online]. Available: <https://proceedings.mlr.press/v139/radford21a.html>.
- [48] L. Ouyang, J. Wu, X. Jiang, *et al.*, “Training language models to follow instructions with human feedback,” in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35, Curran Associates, Inc., 2022, pp. 27730–27744. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf).
- [49] O. AI, : A. Young, *et al.*, *Yi: Open foundation models by 01.ai*, 2024. arXiv: 2403.04652 [cs.CL].
- [50] N. Shazeer, *Glu variants improve transformer*, 2020. arXiv: 2002.05202 [cs.LG].
- [51] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, *Roformer: Enhanced transformer with rotary position embedding*, 2023. arXiv: 2104.09864 [cs.CL].
- [52] J. E. Ramos, “Using tf-idf to determine word relevance in document queries,” 2003. [Online]. Available: <https://api.semanticscholar.org/CorpusID:14638345>.
- [53] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang, *Towards general text embeddings with multi-stage contrastive learning*, 2023. arXiv: 2308.03281 [cs.CL].
- [54] G. Izacard, M. Caron, L. Hosseini, *et al.*, “Unsupervised dense information retrieval with contrastive learning,” *Transactions on Machine Learning Research*,

- 2022, ISSN: 2835-8856. [Online]. Available: <https://openreview.net/forum?id=jKN1pXi7b0>.
- [55] V. Karpukhin, B. Oguz, S. Min, *et al.*, “Dense passage retrieval for open-domain question answering,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. [Online]. Available: <https://aclanthology.org/2020.emnlp-main.550>.
- [56] C. Crawl, *Common crawl: Open repository of web crawl data*. [Online]. Available: <https://commoncrawl.org/>.
- [57] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008. [Online]. Available: <http://www-csli.stanford.edu/~hinrich/information-retrieval-book.html>.
- [58] K. A. Hambarde and H. Proença, “Information retrieval: Recent advances and beyond,” *IEEE Access*, vol. 11, pp. 76 581–76 604, 2023, ISSN: 2169-3536. DOI: 10.1109/access.2023.3295776. [Online]. Available: <http://dx.doi.org/10.1109/ACCESS.2023.3295776>.
- [59] A. Habibi Lashkari, F. Mahdavi, and V. Ghomi, “A boolean model in information retrieval for search engines,” *Information Management and Engineering, International Conference on*, vol. 0, pp. 385–389, Jan. 2009. DOI: 10.1109/ICIME.2009.101.
- [60] J. Han, M. Kamber, and J. Pei, “Getting to know your data,” in *Data Mining (Third Edition)*, ser. The Morgan Kaufmann Series in Data Management Systems, J. Han, M. Kamber, and J. Pei, Eds., Third, Morgan Kaufmann, 2012, pp. 39–82. DOI: 10.1016/B978-0-12-381479-1.00002-2. [Online]. Available: <https://doi.org/10.1016/B978-0-12-381479-1.00002-2>.
- [61] J. Tabak, *Geometry: The Language of Space and Form* (Facts on File math library). Facts On File, Incorporated, 2014, ISBN: 978-0-8160-6876-0. [Online]. Available: <https://books.google.se/books?id=r0HuPiexnYwC>.
- [62] N. Fuhr, “Probabilistic models in information retrieval,” *Computer Journal*, vol. 35, Oct. 2000. DOI: 10.1093/comjnl/35.3.243.
- [63] P. Lewis, E. Perez, A. Piktus, *et al.*, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 9459–9474. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf).
- [64] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, *Large language models struggle to learn long-tail knowledge*, 2023. arXiv: 2211.08411 [cs.CL].
- [65] O. Ram, Y. Levine, I. Dalmedigos, *et al.*, “In-context retrieval-augmented language models,” *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1316–1331, 2023. DOI: 10.1162/tac1\_a\_00605. [Online]. Available: <https://aclanthology.org/2023.tac1-1.75>.
- [66] R. Teja, *Evaluating the ideal chunk size for a rag system using llamaindex*, Accessed: 2024-05-15, 2023. [Online]. Available: <https://www.llamaindex>.

- ai/blog/evaluating-the-ideal-chunk-size-for-a-rag-system-using-llamaindex-6207e5d3fec5.
- [67] Y. Wang, N. Lipka, R. A. Rossi, A. Siu, R. Zhang, and T. Derr, *Knowledge graph prompting for multi-document question answering*, 2023. arXiv: 2308.11730 [cs.CL].
- [68] S. Barnett, S. Kurniawan, S. Thudumu, Z. Brannelly, and M. Abdelrazek, *Seven failure points when engineering a retrieval augmented generation system*, 2024. arXiv: 2401.05856 [cs.SE].
- [69] R. Jagerman, H. Zhuang, Z. Qin, X. Wang, and M. Bendersky, *Query expansion by prompting large language models*, 2023. arXiv: 2305.03653 [cs.IR].
- [70] S. Robertson and H. Zaragoza, “The probabilistic relevance framework: Bm25 and beyond,” *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, Apr. 2009, ISSN: 1554-0669. DOI: 10.1561/15000000019. [Online]. Available: <https://doi.org/10.1561/15000000019>.
- [71] Y. Levine, I. Dalmedigos, O. Ram, *et al.*, *Standing on the shoulders of giant frozen language models*, 2022. arXiv: 2204.10019 [cs.CL].
- [72] S. Pawar, S. M. T. I. Tonmoy, S. M. M. Zaman, V. Jain, A. Chadha, and A. Das, *The what, why, and how of context length extension techniques in large language models – a detailed survey*, 2024. arXiv: 2401.07872 [cs.CL].
- [73] P. Zhang, Z. Liu, S. Xiao, N. Shao, Q. Ye, and Z. Dou, *Soaring from 4k to 400k: Extending llm’s context with activation beacon*, 2024. arXiv: 2401.03462 [cs.CL].
- [74] J. Hsia, A. Shaikh, Z. Wang, and G. Neubig, *Ragged: Towards informed design of retrieval augmented generation systems*, 2024. arXiv: 2403.09040 [cs.CL].
- [75] N. F. Liu, K. Lin, J. Hewitt, *et al.*, “Lost in the middle: How language models use long contexts,” *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:259360665>.
- [76] *Strategy: Test changes systematically*, Accessed: 2024-05-13, 2024. [Online]. Available: <https://platform.openai.com/docs/guides/prompt-engineering/strategy-test-changes-systematically>.
- [77] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, *A systematic survey of prompt engineering in large language models: Techniques and applications*, 2024. arXiv: 2402.07927 [cs.AI].
- [78] X. Amatriain, *Prompt design and engineering: Introduction and advanced methods*, 2024. arXiv: 2401.14423 [cs.SE].
- [79] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, “Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 09, pp. 2251–2265, Sep. 2019, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2018.2857768.
- [80] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, “A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities,” *ACM Comput. Surv.*, vol. 55, no. 13s, Jul. 2023, ISSN: 0360-0300. DOI: 10.1145/3582688. [Online]. Available: <https://doi.org/10.1145/3582688>.
- [81] S. Es, J. James, L. Espinosa Anke, and S. Schockaert, “RAGAs: Automated evaluation of retrieval augmented generation,” in *Proceedings of the 18th*

- 
- Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, N. Aletras and O. De Clercq, Eds., St. Julians, Malta: Association for Computational Linguistics, Mar. 2024, pp. 150–158. [Online]. Available: <https://aclanthology.org/2024.eacl-demo.16>.
- [82] A. Celikyilmaz, E. Clark, and J. Gao, *Evaluation of text generation: A survey*, 2021. arXiv: 2006.14799 [cs.CL].
- [83] D. Muhlgaay, O. Ram, I. Magar, *et al.*, “Generating benchmarks for factuality evaluation of language models,” in *Conference of the European Chapter of the Association for Computational Linguistics*, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:259847758>.
- [84] S. Y. Feng, V. Khetan, B. Sacaleanu, A. Gershman, and E. Hovy, “CHARD: Clinical health-aware reasoning across dimensions for text generation models,” in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, A. Vlachos and I. Augenstein, Eds., Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 313–327. DOI: 10.18653/v1/2023.eacl-main.24. [Online]. Available: <https://aclanthology.org/2023.eacl-main.24>.
- [85] K. Singhal, S. Azizi, T. Tu, *et al.*, *Large language models encode clinical knowledge*, 2022. arXiv: 2212.13138 [cs.CL].
- [86] Z. Guo, R. Jin, C. Liu, *et al.*, *Evaluating large language models: A comprehensive survey*, 2023. arXiv: 2310.19736 [cs.CL].
- [87] “Langchain.” Accessed: 2024-05-15. (2024), [Online]. Available: [https://python.langchain.com/v0.1/docs/get\\_started/introduction/](https://python.langchain.com/v0.1/docs/get_started/introduction/).
- [88] “Mteb: Massive text embedding benchmark.” Accessed: 2024-05-15. (2024), [Online]. Available: <https://huggingface.co/blog/mteb>.
- [89] “Weaviate.” Accessed: 2024-05-15. (2024), [Online]. Available: <https://weaviate.io/developers/weaviate>.



# A

## Retrieval Prompt Template

"" You are an AI Software Developer assistant who is an expert and deeply knowledgeable in AUTOSAR (AUTomotive Open System ARchitecture), your role is to provide accurate, AUTOSAR 4.0-specific responses. ALWAYS UTILIZE the context provided below to answer questions: \n

{formatted\_context}

\n Where applicable, INCLUDE code snippets of ARXML and examples to clarify and illustrate your responses. If the context lacks the necessary information, clearly state "Insufficient information available in retrieved context" and REFRAIN from a speculative answer. Your commitment to these instructions is crucial for maintaining information integrity. ""

---



# B

## Query Generation Prompt Template

"" As a Software Developer AI assistant with expertise in AUTOSAR (AUTomotive Open System ARchitecture), you are tasked with generating questions that delve into the specifics of the given context.

Given the context: \n {context}

\n craft a question that thoroughly explores the subject matter of the context. Your question should aim to provoke thoughtful consideration or detailed explanation related to AUTOSAR, leveraging your deep knowledge in the field. Ensure that the question is directly relevant to the provided context, allowing for an insightful exploration of AUTOSAR-related topics.

Be very strict and only provide the question as the output and nothing else! Do not provide any output such as "To ensure strict adherence to the guidelines, here is the question that thoroughly explores the subject matter of the provided context:" or "In the context provided". ""

---



# C

## Survey

**Question: How to implement E2E (End-to-End) protect and security for a signal group?**

**Question: How to configure and set up an diagnostic messages routine and process step-by-step?**

**Question: How to configure and handle CAN Flexray data message, both in transmission and reception of a single frame? Please guide me in the process.**

**Question: How can the SecOC module integrate with PDU to enhance security? Provide me implementation details and the steps to be taken!**

**Question: What steps needs to be taken to implement CAN-FD considering enhancement of data transmission rates and payload capacity?**

**Does the response correctly address the question?**

*Please comment on its accuracy.*

**Response 1:**

Ditt svar

**Response 2:**

Ditt svar

**Response 3:**

Ditt svar

**How logically coherent do you find the reasoning in the response?**

*Please comment on its clarity and logical flow.*

**Response 1:**

Ditt svar \_\_\_\_\_

**Response 2:**

Ditt svar \_\_\_\_\_

**Response 3:**

Ditt svar \_\_\_\_\_

**Does the response contain any irrelevant information or inaccuracies?**

*Please provide a short comment on these aspects. If time, point out a specific example.*

**Response 1:**

Ditt svar \_\_\_\_\_

**Response 2:**

Ditt svar \_\_\_\_\_

**Response 3:**

Ditt svar \_\_\_\_\_

**Which response do you find the most compelling, and which do you find the least compelling?**

*Please provide a short comment on why.*

Ditt svar \_\_\_\_\_

# D

## Dataset Link

Github