



CHALMERS



# Supporting the generation of engineering analysis reports with Large Language Models

A study of needs and exploration of solutions

Master's thesis within the university programs: Systems, Control and Mechatronics and Product Development

DANIEL SÖDERQVIST, FELIX MARE



MASTER'S THESIS 2024

# Supporting the generation of engineering analysis reports with Large Language Models

A study of needs and exploration of solutions

Daniel Söderqvist  
Felix Mare



**CHALMERS**

Department of Industrial and Materials Science  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg 2024

Supporting the generation of engineering analysis reports with Large Language Models

A study of needs and exploration of solutions

Daniel Söderqvist, Felix Mare

© Daniel Söderqvist, Felix Mare, 2024.

Supervisors: Alejandro Pradas, Najeem Muhammed, GKN Aerospace Sweden

Examiner: Ola Isaksson, Department of Industrial and Materials Science

Master's Thesis 2024

Department of Industrial and Materials Science

Chalmers University of Technology

SE-412 96 Gothenburg

Telefon +46 31 772 1000

Cover image: AI-generated image made with the prompt "*Can you generate small animated robot agents representing llms helping an engineer with his report writing and reviewing tasks?*".

Written in L<sup>A</sup>T<sub>E</sub>X

Gothenburg 2024

# ABSTRACT

At GKN Aerospace Sweden, engineers work hard to conduct simulations and analyses to verify the performance of their products. Being an aerospace company, the main product portfolio consists of aero-engine components, space systems and upgrades of military aircraft engines. The outcome of the analysis performed on these types of products needs to be documented in technical *analysis reports*. To go from the generation of a report to the final stamp of approval, 3 roles are involved: Author, reviewer and approver. The process often has long lead times and suffers from many unnecessary loops and iterations back and fourth between the roles. With the rapid rise of AI in the form of Large Language Models, there is an interest in evaluating how this technology can be used to aid the engineers in this process and reduce lead time. This is the main aim of this study. An important factor is that due to confidentiality, data from the company cannot be transferred to unauthorized LLM providers. This study conducts a thorough interview study with engineers responsible for the different parts of the process. A qualitative thematic analysis of the interview transcripts then resulted in a comprehensive process problem analysis clearly mapping the issues within. Literature research was carried out providing a wide overview of the recent advancements in the field of Large Language Models and inspiring implementations. Using the problem analysis and literature, several design criteria were defined and explored based on how the process can be supported using LLMs. Software was developed using the Langchain framework in Python, resulting in 12 different concepts to support the process. The outcome shows that LLM has the potential to generate reviews of the engineering analysis reports, producing promising results in five test cases created regarding the topics language, content, references and consistency. However, the study finds that in the cases where the concepts found a high proportion of report errors, there were also problems with the production of a lot of false or untruthful feedback. Further demonstrations of generating or retrieving parts of a report were developed and are presented as proof-of-concept. Lastly, a prototype of a user interface visualises how GKN Aerospace engineers might interact with the developed functionality. The study strongly indicates that GKN Aerospace can benefit to a great extent from such solutions. All code implementations can be found at [https://github.com/DaniSode/Master\\_Thesis](https://github.com/DaniSode/Master_Thesis).

Keywords: Large Language Models, LLM, Engineering, Support, Reasoning, Reflection, Multi-agent



# Preface

We extend our largest gratitude to Alejandro Pradas, Najeem Muhammed and Ola Isaksson for their invaluable guidance and supervision throughout this thesis. A special thanks to Alex for taking the initiative to engage in social activities outside of work hours such as playing floor-ball and badminton. Thank you also for the nice dinner together. Activities like these helped keep the spirit high throughout the thesis. Thanks also to Atharva Naik for great collaboration during the 20 weeks of work and for his opposition and comments during the final presentation. A big thank you to Rikard Nedar for the warm welcome, help in providing a good space to work at the office and for the opportunities to meet and speak to the engineers at the PVE: Solid Mechanics department at GKN Aerospace's facilities in Trollhättan. Lastly, thanks to every engineer who participated in the interview study and thereby helped creating this thesis.

Daniel Söderqvist & Felix Mare, Gothenburg, June 2024



# Contents

<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xv</b>
<b>List of Prompts</b>	<b>xvii</b>
<b>List of Outputs</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Aim . . . . .	4
1.3 Research questions . . . . .	4
1.4 Limitations . . . . .	4
1.5 Ethical considerations . . . . .	5
1.6 Technical background . . . . .	7
1.6.1 LangChain . . . . .	7
1.6.1.1 Chat models . . . . .	7
1.6.1.2 Agents and Multi-Agent Systems . . . . .	8
1.6.1.3 LangSmith . . . . .	8
1.6.1.4 LangGraph . . . . .	9
1.6.2 Retrieval-Augmented Generation . . . . .	9
<b>2 Methodology</b>	<b>11</b>
2.1 Data collection . . . . .	11
2.1.1 Interviews . . . . .	12
2.1.2 Literature research . . . . .	14
2.1.3 Review of company documents . . . . .	16
2.2 Data analysis . . . . .	16
2.2.1 Preliminary interview coding . . . . .	16
2.2.2 Thematic analysis of the process problems . . . . .	17
2.2.3 Analysis of literature . . . . .	18
2.3 Concept development and evaluation . . . . .	18
2.3.1 Definition of design criteria . . . . .	19
2.3.2 Development strategy . . . . .	19
2.3.2.1 Dummy report . . . . .	21
2.3.2.2 Guidelines . . . . .	21

<b>3</b>	<b>Results</b>	<b>23</b>
3.1	Interview results . . . . .	23
3.1.1	Analysis of process problems . . . . .	23
3.1.1.1	Writer perspective issues . . . . .	23
3.1.1.2	Reviewer perspective issues . . . . .	25
3.1.1.3	General issues . . . . .	26
3.1.1.4	Not process specific issues . . . . .	29
3.2	Literature findings . . . . .	30
<b>4</b>	<b>Concept development and testing</b>	<b>35</b>
4.1	Design criteria to evaluate . . . . .	35
4.2	Measuring performance . . . . .	36
4.2.1	Criteria 1-5 . . . . .	36
4.2.2	Criteria 6-7 . . . . .	37
4.2.3	Criteria 8 . . . . .	37
4.3	Concepts . . . . .	38
4.3.1	Parsing the analysis reports . . . . .	38
4.3.2	Design criteria: Review language . . . . .	38
4.3.2.1	Language 1 . . . . .	39
4.3.2.2	Language 2 . . . . .	41
4.3.2.3	Language 3 . . . . .	42
4.3.2.4	Comparison of approaches . . . . .	43
4.3.3	Design criteria: Review content . . . . .	45
4.3.3.1	Content 1 . . . . .	45
4.3.3.2	Content 2 . . . . .	47
4.3.3.3	Comparison of approaches . . . . .	50
4.3.4	Design criteria: Review reference usage . . . . .	52
4.3.4.1	Correct reference 1 . . . . .	52
4.3.4.2	Correct reference 2 . . . . .	57
4.3.4.3	Comparison of approaches . . . . .	58
4.3.5	Design criteria: Review report consistency . . . . .	59
4.3.5.1	Consistency 1 . . . . .	59
4.3.6	Design criteria: Review missing references . . . . .	63
4.3.6.1	Check for missing reference 1 . . . . .	63
4.3.6.2	Check for missing reference 2 . . . . .	65
4.3.6.3	Comparison of approaches . . . . .	67
4.3.7	Design criteria: Retrieve old reports . . . . .	69
4.3.7.1	RAG . . . . .	69
4.3.8	Design criteria: Generate new parts of report . . . . .	70
4.3.8.1	Generate . . . . .	70
4.3.9	Design criteria: Communicate to user . . . . .	73
4.3.10	Summary of performance . . . . .	75
<b>5</b>	<b>Discussion</b>	<b>77</b>
5.1	The interview study . . . . .	77
5.2	Findings in literature . . . . .	78
5.3	LLM-based aid in the process . . . . .	79

---

5.3.1	Scripting vs LLM . . . . .	79
5.4	The developed concepts . . . . .	79
5.4.1	Performance on the defined measures . . . . .	79
5.4.2	Human evaluation of model output . . . . .	80
5.4.3	Added value of the solutions . . . . .	80
5.4.4	The dummy report used . . . . .	81
5.4.5	Dummy references . . . . .	81
5.4.6	Testing process . . . . .	82
5.4.7	Comparison to human reviewing . . . . .	82
5.4.8	Ideas not implemented . . . . .	82
5.5	One type of report and process . . . . .	83
5.6	State-of-the-art vs open source models . . . . .	84
5.6.1	Implementation at GKN Aerospace . . . . .	84
<b>6</b>	<b>Conclusion</b>	<b>85</b>
6.1	Recommendations and further research . . . . .	86
	<b>Bibliography</b>	<b>87</b>
<b>A</b>	<b>Data collection</b>	<b>I</b>
A.1	English interview guide . . . . .	I
<b>B</b>	<b>Concept development documents</b>	<b>III</b>
B.1	Dummy Report . . . . .	III
B.2	Language guidelines . . . . .	IV
B.3	Content guidelines . . . . .	VI
<b>C</b>	<b>Test cases and results</b>	<b>IX</b>
C.1	Design criteria: Review language . . . . .	IX
C.2	Design criteria: Review content . . . . .	XI
C.3	Design criteria: Review correct reference usage . . . . .	XIII
C.4	Design criteria: Review consistency . . . . .	XV
C.5	Design criteria: Review missing references . . . . .	XVII



# List of Figures

1.1	Figure showing a simple example and representation of a graph with its nodes and edges with cyclical steps from the LangGraph library. . . . .	9
1.2	Flowchart representing the functionality of RAG. . . . .	10
2.1	Schematic of the methodology . . . . .	11
2.2	Schematic of the development process workflow . . . . .	20
4.1	Average results of Language 2 . . . . .	42
4.2	Average results of Language 3 . . . . .	43
4.3	Results on DC1 using GPT-4 . . . . .	44
4.4	Token usage on DC1 using GPT-4 . . . . .	44
4.5	Average results of Content 1. . . . .	47
4.6	A schematic of the approach in Content 2. . . . .	48
4.7	Results of testing Content 2. . . . .	50
4.8	The final results on DC2 using GPT-4. . . . .	51
4.9	Token usage to assess the full report for both content concepts. . . . .	51
4.10	Simplified schematic of the approach in correct reference 1 . . . . .	53
4.11	The docstring of the reference retriever tool. . . . .	54
4.12	The final results on DC3 for the two approaches (GPT-4) . . . . .	58
4.13	The token usage required for assessing the full report for the Correct reference concepts with GPT-4 . . . . .	59
4.14	Simplified schematic of the logic in Consistency 1. . . . .	60
4.15	Full results on DC4 for the only concept developed. . . . .	63
4.16	Results for DC6 using GPT-4. . . . .	68
4.17	Token usage to assess the full report for both missing references concepts. . . . .	68
4.18	A prototype of the intended user interface home page . . . . .	74
4.19	A prototype of the report-review page in the user interface. . . . .	75



# List of Tables

3.1	Summary of interview subjects . . . . .	23
4.1	The defined design criteria to evaluate. . . . .	36
4.2	Full performance results for all developed concepts using GPT-4. . . . .	75
B.1	Guidelines for language with a corresponding example for each guideline. . . . .	V
B.2	Guidelines for content for each section of the report. . . . .	VIII
C.1	The test case used to evaluate DC1 . . . . .	IX
C.2	The results from the tests to evaluate DC1 . . . . .	X
C.3	The test case used to evaluate DC2 . . . . .	XI
C.4	The results from the tests to evaluate DC2 . . . . .	XII
C.5	The test case used to evaluate DC3 . . . . .	XIII
C.6	The results from the tests to evaluate DC3 . . . . .	XIV
C.7	The test case used to evaluate DC4 . . . . .	XV
C.8	The results from the tests to evaluate DC4 . . . . .	XVI
C.9	The test case used to evaluate DC5 . . . . .	XVII
C.10	The results from the tests to evaluate DC5 . . . . .	XVIII



# List of Prompts

4.1	System prompt used in Language 1. . . . .	39
4.2	Prompt used to generate output in Language 1 . . . . .	40
4.3	System prompt used in Content 1. . . . .	45
4.4	Generation prompt used in Content 1. . . . .	45
4.5	The reflection system prompt used in Content 2. . . . .	48
4.6	The refine system prompt used in Content 2. . . . .	48
4.7	System prompt used for the first node in the system of Correct refer- ence 1 . . . . .	54
4.8	System prompt used for the final node in the system of Correct refer- ence 1 to compare the report to the references used. . . . .	55
4.9	Input prompt used for the final node in the system of Correct reference 1.	55
4.10	The system prompt used in Consistency 1. . . . .	60
4.11	The input prompt used in Consistency 1 . . . . .	60
4.12	System prompt for the last step of Consistency 1. . . . .	61
4.13	System prompt used for the first missing reference concept. . . . .	64
4.14	Human prompt used for the first missing reference concept. . . . .	64
4.15	System prompt used for the first missing reference concept. . . . .	69
4.16	Input prompt as a query to the LLM, example 1. . . . .	69
4.17	Input prompt as a query to the LLM, example 2. . . . .	70
4.18	Input prompt as a query to the LLM, example 3. . . . .	70
4.19	System message prompt for the generation node. . . . .	71
4.20	System message prompt for the reflection node. . . . .	71
4.21	Human message prompt for both the nodes. . . . .	71
4.22	Example of input prompt to the generate concept. . . . .	71



# List of Outputs

4.1	A truncated output from running Language 1 with Mistral 8x7B, showing three $F_{false}$ .	40
4.2	A truncated output from running Language 1 with GPT-4, showing two $F_{intentional}$ .	41
4.3	A truncated output from running Content 1 with Mistral, showing two $F_{intentional}$ from the requirements section	46
4.4	A truncated output from running Content 1 with Mistral, showing a $F_{intentional}$ about mentioning of the process in the results section	46
4.5	A truncated output from running Content 1 with Mistral, showing two $F_{false}$ given in the Analysis models and Method section	46
4.6	A truncated output from running Content 1 with GPT-4, spotting an error that Mistral did not.	47
4.7	A truncated partial output from the initial generation step in Content 2 with GPT-4, giving two $F_{false}$ when reviewing the Method section.	49
4.8	A truncated output from the reflect step in Content 2 with GPT-4, correctly realizing that the two points of feedback are incorrect.	49
4.9	The first node in correct reference 1 deciding that it does not need to go look for reference content.	54
4.10	Truncated output of reference content retrieval in Correct reference 1.	55
4.11	Truncated output of several $F_{intentional}$ when running Correct reference 1.	57
4.12	Truncated output showing an $F_{false}$ when running Correct reference 1.	57
4.13	Truncated output of several $F_{intentional}$ when running Consistency 1.	61
4.14	Truncated output of Consistency 1 showing undesired behaviour. Compare output 4.13.	61
4.15	Truncated output of Consistency 1 showing confusing answers, resulting in an $F_{false}$ .	62
4.16	Truncated output of Consistency 1 showing confusion about results, resulting in an $F_{false}$ .	62
4.17	A truncated output of the missing reference concept with full report using GPT-4, giving two $F_{false}$ points of feedback.	65
4.18	Output of the missing reference concept with the full report using GPT-4, correctly realizing that the two showed points of feedback are incorrect.	65
4.19	A truncated output of the missing reference concept with chunks using GPT-4, giving two $F_{false}$ points of feedback.	66

4.20	A truncated output of the missing reference concept with chunks using GPT-4, correctly realizing that the two points of feedback are incorrect.	67
4.21	Output corresponding to the query 4.16, example 1. . . . .	69
4.22	Output corresponding to the query 4.17, example 2. . . . .	70
4.23	Output corresponding to the query 4.18, example 3. . . . .	70
4.24	Output from generation node without feedback corresponding to the query 4.22. . . . .	72
4.25	Output from generation node with feedback corresponding to the query 4.22. . . . .	72
4.26	Output from reflection node corresponding to the query 4.21 and the first output from the generation node 4.24. . . . .	73

# 1

## Introduction

In recent years, AI has boomed, finding its way into almost every individual's life. In one way or another, society is more exposed to AI as there are now networks that can handle difficult and complex tasks that would have taken humans years to solve. Whether it is work-related or private, there is research and investigation into how AI can more effectively help solve everything from everyday problems to work-related tasks.

### 1.1 Background

One company that today is working with both time-consuming and complex processes and tasks is GKN Aerospace Sweden (GKN). The aerospace division of GKN is not only active in Sweden, but is a global actor with offices and factories around the world, including Bangalore, India. The manufacturing facility and headquarters for engines are located in Trollhättan where approximately 2000 employees work. This facility provide and develop advanced components for engines in both civil and military aircraft but also for space rockets. Maintenance and spare parts for these engines is also an important activity [1]. GKN is particularly known for its good quality with both high safety class and long-lasting sustainable products. Where they are one of the leading companies in their market [2].

Specifically at GKN in Trollhättan, in the preliminary and certification phases of product development an associated linear static analysis report within the solid mechanics department has to be made. Also, for each change or associated input such as geometry, materials and or loads etc, these analysis reports are required to justify the change. The reports are one of such time-consuming processes and could involve several employees. The roles of the employees involved are more specifically, author who is the main person conducting the analysis and writing the report, reviewer who is the person reviewing and giving comments on the written report and lastly, the approver who is also reviewing but have the final say in if the report can be approved or not. If something is incorrect with the resulting report or if something fails during the process, the request will not be approved and will have to be iterated from the start once again. This means that small changes can result in several months of "unnecessary" processing. More specifically, the verification processes are too long, locking key resources, increasing costs and delaying the product

development process.

As mentioned, the whole report generation process is a time-consuming operation where a critical element is the additional loops. Hypothetically, an example would be if a report generation process needed three loops of iterations and takes about one month in total. Since the writer has to write the report and revise and the reviewer has to repeatedly review the report, an LLM could provide aid and feedback. Hypothetically, this can eliminate two of the iterations as the report goes through the approval process in one iteration and by a rough estimate half the lead time for both the writer and the reviewer. Say that the writer spends roughly 40 % and the reviewer spends 10 % of their time on the report generation process, this would mean (based on 160 hours a month) that the writer saved 32 hours and the reviewer 8 hours, totaling 40 work hours per month per each report. This time could then have been used to start a new analysis or perform some other task. It is worth noting that this is a very rough estimation and may not be true as it is difficult to estimate when resources, analyses and processes differ.

Most documents and reports included in the process have been marked as confidential by GKN. Since customers are both military and civil, the security of information and data within the company's borders is important. If confidential information were to be leaked to actors such as authorities, rivals, etc. the company's credibility and reputation could be damaged. To ensure this security, all employees, as well as those responsible for visitors, must sign a confidentiality agreement that tells them what they can and cannot do with equipment and data from and within the company. At GKN Aerospace there is a strict no export data policy which means that data handled and processed within the company must not be exported outside the boundaries in any shape or form. This policy is one of the regulations that must be approved before any data or area can be entered.

To help in time-consuming processes and tasks that for example employees at GKN faces, AI in the form of large language model (LLM) is introduced. An LLM can handle several different natural language processing (NLP) tasks using deep learning algorithms [3]. NLP describes how a computer or AI can take inputs such as text, speech or even images in a human-like manner, then reason and process the information and finally interact or respond in a similar way. In this way, an LLM can perform tasks such as recognizing, predicting, generating text or other types of data but also translating [4]. To perform these tasks, the LLM consists of transformer models in different formats with different architectures [3]. A transformer is usually composed of two main parts, an encoder and a decoder. These are in turn made up of several different layers and operates with the help of a so-called attention mechanism [5]. These transformer models are trained on a very large amount of data and this training phase is called pre-training. This allows for a great adaptivity and scalability to handle a wide set of tasks and with the help of for example fine-tuning or instruction-tuning, the pre-trained LLM base model can be specified to solve tasks that are unique to only that work area. Such tasks can be e.g. understanding protein structures within the healthcare industry or a question answering assistant for a chatbot within customer service. Therefore the LLM's are not limited to only

handle text-generation [3].

Among the strengths of an LLM is that it has information and data from a large number of sources, which also means that the general knowledge bank is large with the model having expertise in a wide range of subjects [6]. Therefore, the LLM has a great ability to analyze and process text, whether it is to summarize, give opinions or improve the text. Another strength is the ability to create and generate material through a question or explanation [7]. However, this opens up for disadvantages such as the LLM only representing the information that was real at the time of training the model. For example, the GPT-3.5 Turbo model is trained on data up to and including September 2021, which in turn leads to the responses from the model being skewed if used today. There is also a limit on the context window meaning that the LLM only can handle up to a certain amount of data at a time. The LLM is also constrained by not being able to remember and build on its knowledge base with new information from new interactions or tasks. As such, the models tend to be relatively inconsistent where answers can vary from question to question or even sentence to sentence within the same output. This also gives rise to the appearance of invented or false information often referred to as hallucinations. A base model also has limited reasoning capacity, which means that the model sometimes “thinks” in the wrong way and makes mistakes [8].

The LLM models can be further divided into open source and closed source models. An open source model is described as a set of pre-trained basic models where both the source code is accessible and the models can be downloaded free of charge on any local machine, whether for private individuals or companies. This means that the model is easily adapted to specific purposes where the lack of requirements and licenses makes them easy to manage. However, an open-source model places greater demands on the machine that will operate the model as both space and computational power are required to host the stand alone model [9]. Two examples of open-source models are Llama and Mixtral which are provided by Meta and Mistral respectively [10]. A closed model is instead described as the opposite, where neither the source code is available nor the model can be downloaded. This means that the model is only available under special circumstances and with the creator’s approval through, for example, renting [9]. The rental can be done using Application Program Interface (API) which is simply a kind of protocol that enables communication between programs and systems through a structured data transfer process [11]. This in turn means that the model is hosted elsewhere and then the requirement for hardware also reduces. However, all data is sent to and through the model and therefore the creator has access to all that data. An example of a closed-source model is GPT-4 provided by OpenAI [10].

Since LLM has a strength in analyzing and processing text, it is an asset to use when it comes to the report generation process at GKN. Both in terms of peer review of finished reports but also to help generate new material for the reports. This can save both time and unlock resources compared to letting the workers continue to do tasks that an LLM might be able to do. Given the weaknesses of an LLM, the model

still needs to have supervision where the output might need some extra evaluation. However, an LLM model can both provide inspiration and hints for improvements in the process, allowing steps in the time-consuming process to be accelerated or even skipped, which would have made things easier for the engineers. Due to GKN's strict no export data policy and security guidelines, models that can be installed and operated in house (i.e. open source) are of greatest interest. This is because special agreements and extra costs may otherwise arise to resolve all issues with a closed source model.

## 1.2 Aim

The main aim of this thesis is to identify the problems that the engineers face in the analysis report generation process at GKN Aerospace and explore LLMs' capabilities and limitations in supporting it. More specifically, it will investigate how the behaviour of LLMs can be utilized to increase the efficiency of the process, and thereby reduce lead time. To do this, suitable approaches need to be identified, deployed and evaluated. Several prototypes of such support are expected to be developed and showcase the functionality along with clear indications of performance and value provided.

## 1.3 Research questions

The following three research questions (RQ 1-3) will be answered as a result of the project:

1. What are the challenges that engineers at GKN Aerospace face today that delay the approval of analysis reports?
2. What are different LLM-based architectures suitable for improving the analysis report generation process?
3. How could LLM based solutions support the engineering analysis activities considering capabilities and limitations?

## 1.4 Limitations

In this research, it is important to recognize the limitations as it provides a context for the study's findings and ensures a transparent discussion of its scope and applicability. These limitations do not undermine the value of the research but rather highlight areas for future research. The limitations of this paper include the following:

- **Timeframe:** The total time to carry out the research is 20 weeks.
- **Modality:** Only consider the textual modality.

- **Budget:** 10,000 SEK are provided for LLM services.
- **Computational Power:** No additional access to more powerful computing performance was provided.
- **Department and analysis:** Only consider limit Strength analysis of a civil aero-engine component at the Solid Mechanics department.
- **Parts of process:** Only consider steps in the analysis report generation process that succeed the conducting of the analysis itself.
- **No export policy:** Due to a No Export Policy and security classification of documents, such data must not be used without declassification.

## 1.5 Ethical considerations

The exponential rise of AI seen in recent years calls for a profound ethical analysis to discuss how its implementation impacts society and the lives of all individuals. LLMs have tremendous capabilities, many of which are likely yet to be discovered. LLM tools such as ChatGPT facilitate everyday life for many people. However, the potential for LLM-based systems to cause harm must not be underestimated. Therefore, they must be developed and used responsibly while being aware of negative effects. All in order to continue to use and develop the resources of LLMs in a sustainable and ethical way. It is crucial to understand that any information that is fed to an LLM may end up in the hands of its provider. In the context of developing an LLM-based system within the premises of a company such as GKN, this must be taken into account at all times to not break export data policies. Some commonly discussed ethical aspects of LLM-based implementation and use include:

- **Generation of harmful content:** As LLMs are trained on the internet, there is a difficulty in sorting out which information is good and which could potentially contain harmful or discriminatory content. There are methods to filter out toxic data such as guardrail models or reinforcement learning with human feedback. But even if there are methods, the internet is a huge source of information and it is important to keep in mind that the models are not flawless. This means that toxicity content in the output particularly against protected groups may exist [12]. Another aspect is whether the generated data may be used for e.g. military purposes such as informing about protected places or people's location but also through instructions or strategies to do harm [13].
- **Hallucinations and production of incorrect information:** Why hallucinations occur can be due to a number of different reasons, such as overfitting, which means that an LLM follows the training data too well and is unable to predict responses that are outside that information domain. The causes can be many, but LLMs are complex and what causes hallucinations is sometimes

unclear or difficult to find out. Some ethical aspects worth considering when it comes to hallucinations are, for example, where privacy has been violated by the LLM leaking personal information. Since an LLM can produce human-like content, there is a risk that inaccurate or fabricated data is disseminated, which in turn can affect or damage socio-cultural, economic or political forums [14]. The fabricated data can also be used to deceive in order to exploit and cause harm through, for example, theft.

Additionally, as the generated content is often both credible and convincing, it is important not to rely solely on what the LLM generates and rather consider it as assistance and guidance. Being source-critical and well-read on the subject makes it easier to determine whether the information from the produced content is reliable or not. In all cases, there needs to be a final layer of human evaluation to critically evaluate the output of the LLM.

- **Economical aspects:** Today, an LLM can generate new material in the form of text, images or even music with a quality that matches the work of an ordinary human. However, it should be noted that the LLM can generate material that is similar to material that has already been generated before, i.e. base its work on already created material. The difference, however, is that the LLM usually does this generation more efficiently and less time-consuming than the human. This in turn can lead to companies wanting to take advantage of this efficiency and allows them to make a lot of money from the opportunity. The downside for workers is that jobs may be lost and most people may suffer job losses. Such a change in the labor market can also lead to a disruption of the economy in terms of wages, among other things. Industries where this is particularly evident include healthcare, finance, manufacturing, entertainment and retail. However, LLMs can also create job opportunities as these tools need maintenance, oversight and improvement only that the jobs may shift from one industry to another [15].
- **Environmental aspects:** There are a number of environmental aspects that are important to keep in mind when working with LLMs. The primary component that works in the operation of LLMs is a graphics processing unit (GPU) and given the enormous data and all the parameters to be calculated in an LLM, a large set of these GPUs is required. To begin with, it takes both energy and resources to produce the chips that sit in all the GPUs. Training an LLM then in turn requires enormous computational power and thus resources in the form of electrical energy but also to cool down the large sets of components that quickly get hot during heavy operations [16]. The amount of carbon dioxide emissions that a little more than 3 round-trips between New York and San Francisco emits was required to train GPT-3.5 in energy equivalence according to [17].

The rapidly growing market of LLMs and the large impacts they have on the environment means that a concern is growing among several major actors [16]. According to [18], 5-50 prompt calls are equivalent to about half a liter of water

in resources, which means that a higher awareness of how the LLM technology is used in society is required [16]. Because the environmental aspects are of great importance, research is being conducted into the possibility of reducing the impacts. Among other things, solutions such as power-capping GPUs and early stopping when the models are not performing, have been shown to reduce energy consumption significantly without disrupting performance [19]. Research also shows the potential of so-called lightweight models that show an equally high degree of performance but require a significantly less proportion of energy consumption [20].

- **Open-source models:** An open-source LLM has its advantages in terms of being installed locally and operated in house. With greater freedom in terms of the possibilities of LLMs also risks arise. An open-source LLM is more difficult to regulate and restrict than a closed-source LLM. Therefore, an open-source model can be more easily exploited in terms of being created, deployed and used for harm [21]. It is also important to note that even if the model is open-source, the data on which it is trained upon not always is. An open-source model may even be a fine-tuned version of a larger closed-source model. This also poses risks as the model indirectly represents the data it is trained on [22].

## 1.6 Technical background

The technical background goes into more detail on concepts and functionalities that are referred to throughout the report. The technical background is intended to give the reader a good understanding of the topic in question and the possibility to follow, for example, methodological descriptions or discussions. The section is further divided into subsections, each describing one or a group of different concepts.

### 1.6.1 LangChain

An open-source framework that has been widely used in this research is LangChain. LangChain can be seen as a repository for LLM-based applications and/or functionalities. The idea is that software developers are able to share ideas and inspiration with others through this framework. Langchain has easily accessible packages that can be easily installed to different code platforms and supports several different programming languages. Instead of developing all the functionality and ideas, Langchain can therefore be woven into the own ideas and code to save both time and speed up development [23].

#### 1.6.1.1 Chat models

A prompt template can be seen as an instruction that the LLM uses to interpret and analyze the input text to the model. In LangChain there are a number of different predefined recipes for different sets of these templates. PromptTemplate is a simple template containing the most necessary information or message. Chat-PromptTemplate is a more sophisticated template that has the added functionality

of assigning roles to different instructions or information [24]. Different roles are e.g. SystemMessage which in principle is used as priming for what basic behavior desired from the LLM. HumanMessage represents a human and what it communicates during an interaction with the LLM. AIMessage represents an AI, thus the AI's response to the interaction [25]. All to make it easier for the LLM to distinguish between who is saying what. These two examples of recipes are just a few of several but the user is also given the opportunity to be able to make customizable prompt templates according to the need. Similarly, it is possible to use predefined recipes for how the output should be parsed and with the possibility of being customized [26].

### 1.6.1.2 Agents and Multi-Agent Systems

Within LangChain there is the concept of Agents which refers to LLM-driven models that are able to determine and perform actions. The agent should, according to its own decisions and reasoning based on some type of logic, be able to perform tasks to the best of its ability driven by the LLM. An Agent is customizable and can be equipped with different tools that refer to different types of functionalities such as google search, running a special script etc. An agent can also be equipped with memory of some kind so that it remembers which tasks have been performed or what reasoning has taken place to broaden the contextual understanding. With the help of the LLM, the agent also has the opportunity to observe the outcome of a tool use for example and can choose to redo the process in a different way if the result is not as desired. The concept opens up for capabilities such as a great adaptability that instead of hardcoding in which order or in which way a functionality should be executed, an Agent decides which can then change methodology or input for the desired result [27].

Multi-Agent Systems (MAS) are systems composed of several of the described agents. These agents can or are given the possibility to interact and cooperate to solve given tasks. By adapting and specializing Agents to different settings, one can then take advantage of their different roles to achieve the desired result. MAS open up a number of possibilities such as great flexibility in the execution of tasks but also when it comes to scalability as the system can be equipped with, for example, more agents. Robustness and reliability also tend to increase through multi-agent systems [28].

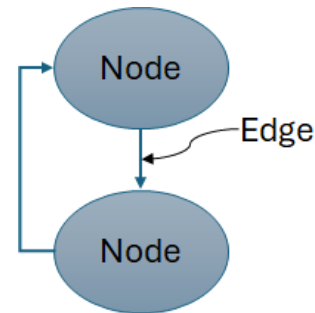
### 1.6.1.3 LangSmith

One functionality within the Langchain framework is LangSmith. LangSmith is a platform that helps developers using LLMs to track the cost, latency or the overall quality of their usage. Through API, there is a smooth communication between the application and each individual project. Through simple settings, multiple projects can be set up in the app and they can be tracked individually or together, which is up to the user. Projects in LangSmith can also be shared between several users so that a smooth opportunity for collaboration is offered. Each run of an LLM can then also be monitored in terms of input to the model, how it processes data, what

data the model sees and the output from the model. If more complex scripts are used, each step in the process can be monitored according to the same principle [29].

#### 1.6.1.4 LangGraph

One library within the Langchain framework is LangGraph. LangGraph is based on using nodes and edges to build a network called a graph. The nodes can mean different functionalities and or models that are then connected with the edges. This allows the user to smoothly coordinate multiple chains of functionalities over mainly cyclical steps. When executing a graph, it all starts with the first node being executed and an output is produced, also called state. The state is then passed on to the next node through the associated edge that runs and updates the state. This process can be looped and or customized to the user's needs using for example conditions. Each step in the process is saved in a so-called MessageGraph that can be easily accessed to overview intermediate steps [30]. Figure 1.1 shows a simple graph where the network is made up of two nodes that cyclically run according to some condition. In this example, the first node represents an LLM that writes a story. The second node is an LLM that then gives feedback on the story. Then the state is sent back to the first node which gets the chance to rewrite the story using the feedback. This can be further looped according to the user's preference.



**Figure 1.1:** Figure showing a simple example and representation of a graph with its nodes and edges with cyclical steps from the LangGraph library.

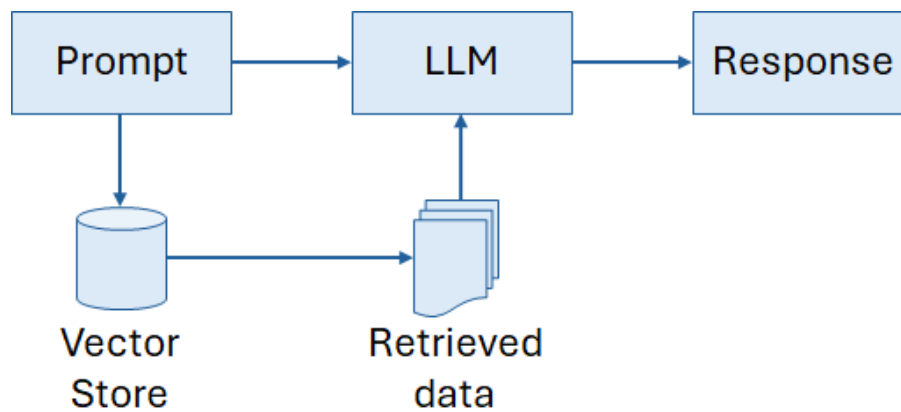
### 1.6.2 Retrieval-Augmented Generation

An LLM has a wide range of knowledge with enormous expertise derived from all the data on which it is trained. However, there are limitations, such as when the model needs to have access to current data or some case-specific material. The LLM then bases its output and answers on the knowledge it has but which in those cases can be completely irrelevant and unhelpful. This is where Retrieval-Augmented Generation (RAG) comes into play and is a method based on the fact that the current or case-specific data that was missing in previous cases should be in a digital library that the model has access to and can use to find the relevant information [31].

The digital library is based on the technology of vector embeddings, which is a numerical representation of an object in vector form. The object in question can be a paragraph or even individual words, the size depends entirely on the need. It is worth mentioning that the more objects and smaller pieces, the more vectors that represent them exist. Numerical representations in vector form take up less space than the object itself, therefore vector embeddings are a convenient way to reduce and store a large amount of data [32]. The library is then created using developed LangChain packages that construct a kind of storage called vector store or vector database.

Two popular vector store methodologies are Chroma and FAISS. They work in such a way that the text in question is divided into smaller parts or so-called chunks that will then become vector embeddings. Depending on the methodology and choice of embedding functionality (OpenAI offers OpenAIEmbeddings for example), a digital library of vector embeddings is then created that represents the chunked document or documents [33].

With the help of the vector store, a request in the form of a prompt can then be interpreted and relevant documents or parts of documents can be retrieved which together with the LLM can form a basis for the output. Figure 1.2 shows a flowchart of how a simple RAG system works and can be structured.



**Figure 1.2:** Flowchart representing the functionality of RAG.

# 2

## Methodology

The methodology of the problem-solving phase from research questions one to three and the purpose of the report follows a chronological order. To lay a broad foundation and understand what the actual problem for the engineers in their analysis report generation process at GKN is, a data collection in the form of interviews was performed. To further improve the understanding a review of approved analysis reports was made. In order to solve the expressed problems in the report generation process, the potential capabilities but also the disadvantages of LLMs with different existing concepts, were investigated. This investigation was done through additional data collection in the form of literature research. Using the data and understanding, gained from the data collection, the implementation and concept development phase was started. With the deeper understanding provided by the interviews and literature review, a list of criteria, verification methods and an initial concept proposal were created. Subsequently, the implementation phase continued in an agile methodology to arrive at several developed concepts. Figure 2.1 shows the chronological order of the implemented methodology as a flow chart.

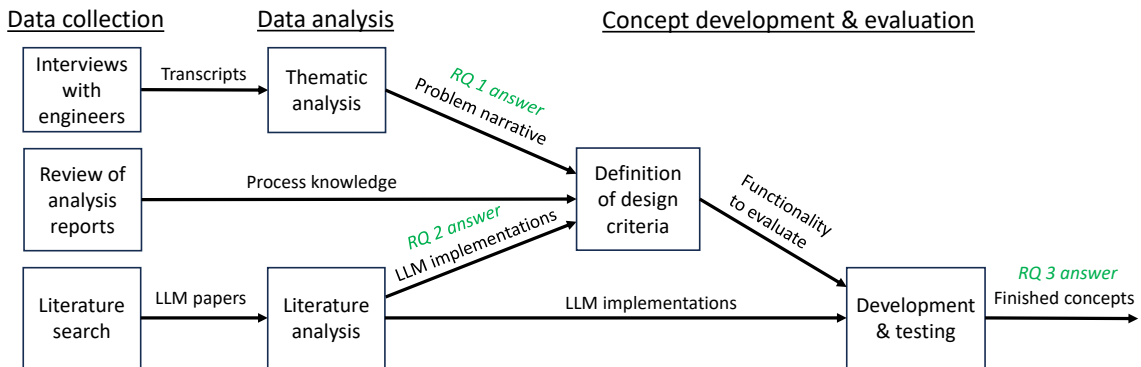


Figure 2.1: Schematic of the methodology

### 2.1 Data collection

In this research paper, three different approaches to data collection were considered: Interviews, literature research and lastly review of company documents. The inter-

views were a primary data collection approach and also in the form of a qualitative approach. The interviews were performed to deep-dive into the existing problems and with the data collected from the literature research, viable solutions were considered. By using primary data and interviews tailored for the specific purpose the validity and reliability of the research improved. The literature research and the review of company data used a secondary data collection approach in the form of a qualitative study. This study was done to lay a broad foundation of existing problems in the researched areas but also to investigate possible solutions to those problems. The interviews in combination with the results from the review of company data helped in answering RQ1 in the form of identifying the different challenges that engineers might face in the analysis report generation process. The literature research helped in answering RQ2 in the form of detecting and investigating different LLM approaches that are suitable to apply in the analysis report generating process, more specifically generating parts of the report and or reviewing the reports. The data collected and processed in the interviews as well as in the review of the company data processes are confidential and cannot under any circumstances be shared or exported outside of the company. All steps in the process were therefore carried out and managed with care and with an awareness of this security risk.

### 2.1.1 Interviews

As stated in [34], interviewing is a powerful tool suitable for capturing complex and subtle problems related to human experience. Interviews as a method for data collection offer several advantages. Because of the direct continuous interaction between the interviewer and the interviewee, the information produced captures the topic with depth and detail. The information can also be assessed for relevance directly when it is produced. Interviews are also flexible as it is easy to adjust the format on the fly. Furthermore, interviews usually have a high response rate when scheduled at convenient times for the interviewees [34]. These are the main reasons why this method was used to collect data to answer RQ1. The analysis report generation process at GKN Aerospace is complex. Because of this, interviews were selected to understand its problems because of the advantages given above.

A disadvantage of using interviews is that the response format is not standardized, this requires a large effort to structure and consolidate the data in a way that is easy to analyse and make use of it. The conversational skills of the interviewer might also be a limiting factor in terms of how the topic gets explored. There is also a risk of bias among the interviewees [34].

The population of interest from which information was collected consisted of every employee who is involved in the analysis report generation process of different types of projects at GKN. The main roles in this population are authors, reviewers and approvers. The population sample goal was to have a representation of every role with a ratio of two authors and two reviewers per approver. Subjects that had previous experience with large language models were preferred because of their potential to provide a valuable perspective on how the LLM can improve the process. Potential interview subjects from the population were initially provided by a company

representative. They were then contacted through an internal email stating the purpose of the research along with a suggested time slot for conducting the interview. Additional candidates were also identified by asking the subjects at the end of each interview.

The interview format was semi-structured and based on some characteristics mentioned by Merriam [35]. These include:

- Interview is guided by a list of issues or questions to be explored
- Exact wording and the order of questions are not defined ahead of the interview, allowing the interviewer to adapt to the emerging answers of the respondent.
- Questions are used flexibly

This interview format was selected because of the substantial knowledge gap regarding the analysis report generation process at GKN Aerospace. Not having previous knowledge makes it hard to ask specific and relevant questions. Therefore, a more open and exploratory format was desired. However, some structure was needed to make sure that the interviews did not drift away from gathering answers to RQ1. An interview guide was used and designed slightly differently based on the role of the respondent and their relative experience with the analysis report generation process. As the process progressed, it was also adjusted due to gradually gaining more knowledge about the topic. The language used was either Swedish or English depending on which the subject was most proficient in to ensure efficient communication. The English interview guide used for the process can be found in appendix A.1.

The maximum number of conducted interviews was set at 10. This upper limit was defined to make sure that the interview phase was finite, thereby reducing the risk of it taking too long. If the saturation point was considered reached before this limit, no more interviews were conducted and the project proceeded to the data analysis step. The interviews were conducted in person to increase the engagement of the respondent and for more natural interaction, this increased the chance of reliable answers. They were also audio-recorded upon giving consent. In both cases, field notes were taken simultaneously by one of the interviewers. The length of the interviews was approximately 40 minutes and they were later transcribed preliminary using speech-to-text tools. The transcripts were then adjusted manually to correctly match the audio recording.

Several ethical concerns were considered when conducting the interviews. As discussed by Brinkmann and Kvale in [36], informed consent needs to be obtained from the subjects participating in the study. This means letting the respondents know about the purpose, benefits and potential risks of the research as well as pointing out that participation is voluntary. However, since the interviews were aimed at employees at GKN Aerospace, consent was obtained both from the subjects and their superiors. Another important ethical aspect considered was anonymity, meaning

that no private data that could identify the subjects was reported unless given explicit consent. It also needed to be clear to all parties involved who will have access to the information after the study and for how long. Another major ethical concern was that the information gathered from the interviews could contain confidential content and was therefore treated according to the regulations and guidelines at GKN Aerospace.

Some alternative methods for data collection that were not used are surveys and focus groups. Surveys are most effective when collecting high volumes of standardized data from identical questions. The desired information from GKN Aerospace concerned the problems that cause delays in the analysis report generation process. These were likely not very straightforward and required nuance to describe. Surveys are more suitable when the desired information is rather simple. Focus groups were disregarded because it would be time-consuming to set up at the company given the engineer's busy schedules. It was also preferable to gain insight individually from the subjects to reduce the risk of important experiences being left out in a larger discussion [34]. Observations were disregarded because the report-generation process is time-consuming and it would therefore take a long time to collect useful information.

### 2.1.2 Literature research

To answer mainly RQ2, but also to lay a foundation for answering research question three, a literature research was made. The most used and famous databases are usually not accessible for free to everyone but in the case of this study, the library at Chalmers University of Technology was granting access to the used databases. The literature research was made using the following process structure [37]:

1. **Preparations:** What is the purpose of the research and what are the questions to be investigated? How can we limit the search to our area? Which keywords highlight the topic? What constraints are to be considered? Plan and set aside a good amount of time since literature research can be time demanding.
2. **Introduction to the subject:** Read up on the subject briefly to get a superficial idea of the subject. This can be done through e.g. informative videos, encyclopedias, articles, etc.
3. **Choose appropriate search tools:** Depending on the preparation and the type of literature research, which tools are appropriate to use?
4. **Do the literature research:** According to the tools and preparations for the review, also adapt keywords and constraints as you go along to ensure an acceptable result.
5. **Read and process the found literature:** Is the found material good? Does it answer the questions asked? Who wrote the work? Is the material previously cited?

6. **Evaluate:** Could the questions asked, be answered? Are the results satisfactory? Does the study need to be extended?

Specific constraints that limited the investigation to the desired area were preliminary set to:

- Keywords: A few examples are Large Language Models, Cognitive Abilities, Agents, Reasoning, Capabilities, etc.
- Preferably consider peer-reviewed literature.
- Only consider literature written in English as the translation of literature may skew the content.
- Publication date: Only consider literature that is no older than 3 years.
- Amount of citations: Literature that is cited at least 40-50 times or in case the literature is published recently, relatively many times considering the publication date.

Three articles were given and used as step two in the literature research process describing multi-agent implementations [38][39] and an implementation using reasoning and reflecting within LLM's [40]. Along with the given articles Youtube videos of concepts such as multi-agents, reason and act and prompt engineering, also frameworks such as Langchain, Langgraph and RAG. All to be familiarized and be given an introduction to the subject. The keywords defined were preliminary and were complemented with suitable words along the literature research process. In the literature research, the preliminary choice of tools was mainly through the Library of Chalmers built-in search engine. In this search engine, several databases were used, such as Scopus, Academic Search Index and Arxiv, which increased access to more literature. If the literature was not peer-reviewed, further investigation of the citations was made. If the article was cited relatively many times considering the publication date, this was considered a preliminary sign of trustworthiness. However, since many citations do not always imply a trustworthy source, it was important not to make an assessment based solely on this aspect.

Another tool used was Connected papers where the found and determined "good" articles were used as input to generate and find similar material. Connected papers also provided a good overview of the number of citations together with the path to the article, usually in the Arxiv database. When useful literature was found using Connected Papers, the articles were further investigated using databases such as Scopus. The investigation consisted of finding out whether the article was published in a journal or not, if it was peer-reviewed, etc, all to ensure the reliability and validity of the articles. Approximately 30 articles within the given limitation were found where the abstracts were read and interpreted. This overall examination was then used as a basis for screening the articles. A criterion that weighed heavily in the screening process was whether the literature had associated GitHub repositories shared publicly. The screening process resulted in 11 articles that were further used

in the research and presented in a summarized table as the results of the literature research.

### 2.1.3 Review of company documents

To better understand the generation process of analysis reports the company provided completed reports which were reviewed. More specifically in the form of written, reviewed and approved reports that were accepted according to the regulations to which GKN adheres. The reports provided were both of varying quality, but also provided in different versions from the different steps in the analysis report generation process along with the comments from the reviewers and approvers. This ensured capturing the whole report generation process more in detail and with the varying quality and comments also a wide range of problems. More specific elements that were examined are the overall content of different sections, the structure of the report and how the reference management is handled, etc. Together with the reports, general guidelines on how to write these reports were included. The results of the literature research of the reports contributed to laying a valuable foundation for the design of the interview template, but also in regards of answering RQ1.

The reports described in this section are subject to the confidentiality policy and were therefore handled and processed accordingly. It was therefore important to consider where the reports were sent or downloaded to and who had access to the material. In this research, only the overall structure and general content of each report were considered to build a basic understanding of the analysis process of report generation. Therefore, the specific content of each report did not had to be saved or taken further into processing, which means that the risk of data leakage were reduced. The reports were only read and processed at the workplace or with equipment provided by GKN Aerospace so that the data were not spread outside the described boundaries.

## 2.2 Data analysis

When the data had been collected, it was analyzed to gain an objective understanding of the content and to ensure that no important information was missed. This process aimed to find trends and patterns in the data that might not have been apparent at first. This information then served as input when developing solutions.

### 2.2.1 Preliminary interview coding

To structure the interview data and facilitate further analysis, the interview transcripts were processed through a preliminary round of coding and stored in a table format. A deductive coding strategy was selected because the findings need to be related to the research questions [41]. A pre-determined coding frame was defined based on the questions in the interview guide:

1. Interviewee background

2. Expressed key steps of the process
3. Process problems
4. Potential solutions
5. AI as a help tool

The transcript of each subject was carefully read and features were coded according to the coding-frame above. The features from each subject transcript were then stored in a column of the table. The process was performed digitally, making use of Microsoft Excel to create and manage the collection of codes and features. Working digitally allowed for efficient structuring of the data.

### **2.2.2 Thematic analysis of the process problems**

When dealing with lots of complex interview data, an effective approach to make sense of the content is to conduct a thematic analysis. This method provides a structured way of organizing and categorizing large amounts of qualitative information to identify and build a consensus on different themes that are inherent in the data. The table described in 2.2.1 served as input to carry out the following 6 steps of a thematic analysis of the process problems as mentioned by Braun and Clarke [42]:

1. Surveying the data: Reading through the features of the table while actively observing meanings and patterns to gain an overview of the content. Take notes of potential ideas.
2. Creating and refining codes: Features of interest across the table were further systematically assigned to new codes. This was done through a combined deductive/inductive approach. Since the analysis was driven by the interest in researching the problems of the process, there was some notion of what the codes should be beforehand. This characterises a deductive approach. However, there was also a need for a data-driven approach to properly explore the problems without analytic preconceptions. In the inductive approach, codes were developed by letting them emerge from the data itself through scrutiny. Each code was represented by a colour in Excel for visual clarity. The coding was done over several rounds of iteration and refinement, initially using open coding to loosely and tentatively define codes to further divide the features in the table. Descriptive coding was then used to arrive at the final set of codes, meaning to code according to the basic topic of a feature of the data [41].
3. Collating codes with supporting data: The features of the data were then collated into groups based on their code.

4. Grouping codes into themes: The codes along with their respective features were then grouped into larger themes, making sure that each theme has enough data to support it and that the boundaries between them are clear. This was done to identify the main patterns and meaning of the data.
5. Writing a narrative: The main result of the analysis was then communicated by writing a narrative of the themes and their inherent codes. Supporting quotes were used to back up the claims and ensure the validity of the result.

After the interviews were analysed, the subjects were sent the final narrative of the process problems and asked to choose their three most crucial problems. This was done to establish a relative weighting of the identified problems. The result was then used as a foundation to inspire potential LLM-aid.

### 2.2.3 Analysis of literature

To analyze and determine which literature is credible or not and which can be used further in the research, a quality check was done [43]. Since research question 2 investigates which and whether LLM implementations can be used to help in the report generation process, important parts of the literature were:

1. Which framework or implementation was used and in which application area.
2. The methodology and development of the implementation.
3. Results and evaluation of the implementation.

The quality of the mentioned sections was checked by examining whether the literature clearly emphasized the desired elements. First, what the study was about, i.e. what type of framework was being investigated, what type of problem was being addressed and what application area the solution was intended to be used in. Then a clear description of the methodology for developing the implementation, e.g. through Github, which demonstrated a high level of credibility if there was open source code and a clear description. Finally, a clear presentation of the result of the implementation and how the result was evaluated with both disadvantages and advantages. These elements were compiled for each literature used and presented line by line in a summarized list that clearly highlights the main findings of the literature review. The literature review was then used not only to answer the research question but also as a source of inspiration for the concept development and evaluation phase.

## 2.3 Concept development and evaluation

With all the required insights gained from the preceding work, the development and testing phase was started. The challenges identified were addressed by developing a

set of LLM-based concepts through continuous exploration and refinement. Several performance metrics were defined to evaluate and compare the concepts to each other.

### 2.3.1 Definition of design criteria

First, the scope and boundaries of the development effort were set. The purpose of this step was to create a specification of design criteria for the LLM-based solutions. The design criteria were defined as functional targets driven by the interest to deliver value to the engineers and to evaluate exploring different LLM solutions. The design criteria were defined by following the 3 steps below, as discussed by Ulrich and Eppinger on the topic of opportunity identification in [44]:

1. Generation of ideas: Considering the analysed literature on LLM implementations and the weighted process problems gained from the engineers, several ideas for LLM-based tools were generated in an in-person brainstorming workshop together with an engineer at the company.
2. Screening ideas: The generated ideas were later screened using the holistic judgement of the group, technical feasibility and technical complexity as screening criteria.
3. Developing and selecting promising ideas: Further research was put into the remaining ideas. The remaining uncertainty was resolved to finally pick the ones worthy of investigating further considering the scope and time frame of the project.

The resulting design criteria of these steps were stored in a table format and used as a guide for the remaining development and evaluation phase.

### 2.3.2 Development strategy

The development phase was carried out using the principles of an agile development framework. The main characteristic of this development style is to constantly evaluate the project priorities and re-plan based on the current situation. This is done in small cycles referred to as sprints [45]. This strategy was selected due to the risk of having to adapt to sudden changes in the project such as issues with data, changes in requirements, troubles with code implementation or acquisition of new knowledge. The goal of the development effort was to produce a set of different concepts demonstrating functionality as per the defined design criteria. The planned development tasks and desired concept features were stored in a project backlog which was reevaluated at the end of each development sprint every 1-2 weeks to make sure that the development was kept on track based on the current status.

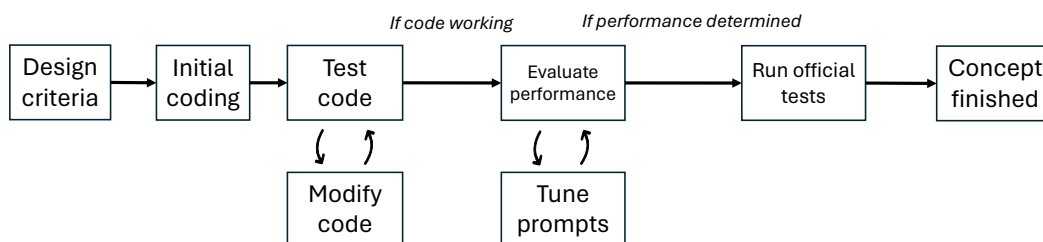
To know how well the concepts worked, the following types of methods were used to evaluate the performance [46]:

## 2. Methodology

---

- Perplexity: A measure of how well the model performs with respect to text prediction. More specifically, how good the generated text is.
- Human evaluation: Involves a human reviewing and evaluating how good the model works and can be with respect to benchmarks such as relevance, fluency, coherence or overall quality, etc.

Building on these methods, additional performance metrics were defined for each type of design criteria taking RQ3 into account. The term concept was defined as an intended solution to one or several design criteria. The boundaries of each concept were articulated as containing a fundamentally different approach to fulfilling the design criteria. They were developed in an exploratory way. First, a design criteria was selected according to the current priority. An idea of a solution fulfilling it was then generated based on literature findings, internal discussion and what could be interesting from an exploration perspective. A preliminary time frame was defined for each development effort. A round of initial coding was then started to create the fundamental parts of the software. The code was then tested and modified in a cycle until working satisfactorily. Upon having functional code, performance on the associated design criteria was evaluated preliminary according to the defined performance metrics. The inherent prompts in the concept were tuned iteratively in an attempt to increase performance. Upon reaching a reasonable indication of the general performance of the approach, the concept was officially tested over several runs and performance on the associated metrics was saved and stored in a table format. If the defined time frame was reached, the process ended prematurely and the development of a new concept started. A flow chart describing the development cycle of a concept is shown below in figure 2.2.



**Figure 2.2:** Schematic of the development process workflow

Upon finishing a concept, the decision of what to implement next was made based on aspects in which the previous could be improved as seen in results from the tests. Ideas for the development of new approaches were found in the analyzed literature. This process was executed several times with the overall development goal of exploring as many design criteria as possible while prioritizing performance on each.

### 2.3.2.1 Dummy report

The majority of documents and reports at GKN Aerospace have a security classification, which places restrictions on how they can be handled. Therefore, all development, evaluation and testing in this research was carried out using fake content. This content only carried a similar structure but without any classified information to simulate a real testing process. In this research, one security declassified report was provided. The report was directly copied from a previously written, reviewed and approved report but with all security classified information deleted or reworked manually by a GKN-employee. Therefore, the report followed both the direct structure of interest but also the writing style used by the authors. All to simulate as good a test case as possible. This in turn increases both the reliability and validity of the results. See Appendix B.1, for the structure and the overall content of each section of the given report.

### 2.3.2.2 Guidelines

Together with the given dummy report, guidelines were obtained for how the authors should write the reports, these guidelines describe how the report should be written in terms of language. In addition, guidelines were obtained for the content of the report, which sections contain which information, which information and which components the sections must contain and how it should be structured. The guidelines given are as follows:

**Language guidelines:** The language guidelines were further divided into subcategories such as tense, passive voice and mentioning of personal and customer's name. The guidelines with a corresponding example for each guideline are shown in Appendix B.2.

**Content guidelines:** The content guidelines were instead of subcategories divided into different guidelines describing how and what content each section of the report should use and consist of. The guidelines for each section are presented in Appendix B.3.



# 3

## Results

This chapter will present the results obtained, processed and analyzed through the methods described in the methodology chapter. The interview subjects will be presented in a table listing their characteristics. This table will be followed by a summary of the mentioned problems sorted according to the different identified themes. The results of the literature review is also presented in a list with the main findings.

### 3.1 Interview results

The sampling from the population of interest resulted in 8 different interview subjects. These are listed below along with their respective experience and role in the process, see table 3.1.

	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	Subject 7	Subject 8
Experience	23 years	3,5 years	12 years	1 year	6 years	23 years	8 years	14 years
Authoring								
Reviewing								
Approving								
LLM experience	Very little	Quite alot	None at all	Has used it a bit	Very little	None at all	Has used it a bit	Quite alot

**Table 3.1:** Summary of interview subjects

#### 3.1.1 Analysis of process problems

The results of the qualitative analysis of the process problems containing themes and sub-themes are presented below.

##### 3.1.1.1 Writer perspective issues

The following theme represents expressed problems from the author's perspective. Sub-themes have first been categorized and presented and these form the basis of the overall theme.

**Writing guidelines:** The existing guidelines and templates may not always be applicable for each type of report, leading to empty sections and confusion regarding

what information should be where and where to start writing. There is also confusion regarding writing style, to what level of detail the authors should express themselves and the type of information that should be included. How to formulate specific keywords is not stated anywhere and it can be hard to know if some part of the report needs to be emphasized. It is also unclear how images and tables should look and authors might create them differently. It would be better if this was standardized. There is an expressed need for a clearer standard of explanation detail and additional guidelines for different sub-genres of reports.

*"Standardize. I think some reports can be standardized, even in the paragraphs, and the way we explain things. "*

- Subject 5

**Generating images, tables and references:** It is time-consuming to generate the images needed for the report since the expected quality is high, especially if the work is done in Ansys Classic. This is therefore problematic in reports that require many images. The creation of tables was also expressed as time-consuming.

*"I feel it is quite time-consuming to take out all the plots and all the tables, I need to make all the stuff and then put it in the report. That's very time-consuming"*

- Subject 7

Creating the file management section of the report is expressed as very slow because you have to list all the related analysis files so that they can be easily tracked later. This process is prone to error as the author might pick the wrong file by mistake.

**Boring/time-consuming to write:** Many subjects expressed that it is very time-consuming to simply write the report, as some of them can reach up to 100 pages. Because of this, it is also considered boring by many and some feel like they just want to get it over with. This in turn leads to lower quality reports. There was an expressed need to speed up the process of writing.

*"One reason why the reports don't turn out so well could be because people don't enjoy writing reports"*

- Subject 4

**Authors workflow:** The quality and delivery of the report are largely dependent on the workflow of the author. Some subjects emphasize the importance of being well aware of everything that needs to be included in the report already when starting the analysis. However, this isn't always the case for the authors which makes the writing process slower since you have to go back again and find the information you need. There is also a risk of forgetting about certain assumptions that were made during the analysis. It was also suggested that an efficient way of working that may not always happen is conducting the analysis and writing in parallel.

*"I would prefer they write in parallel, but some of them just leave the report towards the end. So, that also makes that the report might be delayed"*

- Subject 5

**Old reports:** Some writers choose to pick up and start editing an old previously approved report which already has a bad format without being aware. This can be interpreted as reviewers having different preferences, that guidelines have changed over time or that the old report is not related enough. Furthermore, it is not possible to do a word search in the current file management system to find related old reports, which makes it slow to find what you are looking for. There is an expressed need to find and retrieve good old report content that is related to the current analysis more efficiently.

*"If I or my colleagues write a stiffness report on a common component, it should be quite similar. Yes, that it automatically finds the latest stiffness report that was written."*

- Subject 3

### 3.1.1.2 Reviewer perspective issues

This sub-section addresses the theme, which includes the reviewer's expressed concerns in the process. The theme has then been further categorized using sub-themes.

**Review guidelines:** It was expressed as unclear at what level the review should be applied. For example, how carefully the reviewer should pick up on small things like spelling mistakes, etc. Thus, there are no clear guidelines for how the review should proceed and where the line goes, so the review can be very personally dependent on who reviews and what is approved on which review. Something that also proved to be unclear is where the limit is for which input files that go into the analysis, such as scripts, materials and loads, to be reviewed and to what level of detail they should be reviewed. It is a problem that there is no standardized way to review external files such as scripts.

*"People script in different ways like should I read through a whole script or is it enough that I read through certain parts and which parts in such cases?"*

- Subject 2

Another problematic aspect of the review process when you receive a revised report is that you need to cross-check between the past commented version and the newly revised version each time instead of having comments and changes in the same document.

*"You need to open the copy where you have done the comments, and then the final version, which is stored in team center and cross check. I don't like that either."*

- Subject 5

**Boring and time-consuming to review:** It was mentioned that a difficulty when it comes to reviewing a report was that it was time-consuming and difficult to look through to make sure everything was correct.

*"I basically have to walk the path of the person, the author"*

- Subject 8

Therefore, in this process, there can be difficulties with comprehension at all times and in turn reading through a report can be both time-consuming and quite tedious. It also opens up for simple mistakes such as not checking everything properly and missing some important aspects.

**Hard to review external files:** In the context of the paragraph describing the review guidelines, it was repeatedly expressed that there is no clear line for how deeply to dig into the associated files and references used. In addition, it is expressed that there are often problems related to references, where the references to be used might be prepared with their associated paths for where the files should end up in the end long-term. However, this can get messy if the files used are not ready yet, if they are not in the referenced location or if the files will not even end up in that location. The files can also be hard to find for the reviewer if they are unreleased and in a temporary storage location as the exact path is often missing.

*"then the file might be on nobackup or in the project disk and I don't know where they are"*

- Subject 2

Therefore, a good way of structuring the process can be of great benefit to the reviewer, for example by collecting all external files in an orderly manner and attaching them together with the report in the review process. Related also to the time pressure mentioned by several subjects, it can happen that because of that reason the external files are skipped and they do not get the review they might have needed.

#### 3.1.1.3 General issues

This theme is not linked to any specific person or role in the process but rather addresses general issues in the process mentioned by the interviewees also further categorized into sub-themes.

**Language and formatting:** The reviewers stress the problem that it needs to be clear what the message is in the report and that things are clearly explained, even for people who have not been part of the analysis or who don't have a lot of technical knowledge. Authors may assume that reviewers know more than they do, which leads to some information being left out.

*"I feel that, okay, these points are fairly simple, I need not place it in the report, but it may not be as simple as I think from the reviewer point of view"*

- Subject 7

Some sections of the report were described by reviewers as more important than others such as requirements, purpose and results. They encourage the authors to think about how others might read and understand their text, and that they should aim for short and concise paragraphs that focus on the core message and avoid excessive text. This is a bigger problem for inexperienced authors.

*"A person who is not good at writing reports, who is unsure, often writes a lot of text. So it becomes a nightmare for the reviewer because you don't know what is important to sort out."*

- Subject 6

Report formatting is considered very important in terms of standardized images, tables, captions and fonts to reach the reader. Tables sometimes come with bad formatting, despite there being a formatting guide. They may not be positioned correctly. The same goes for images and their captions, which may have inconsistent positioning.

*"that is mostly the font size or it is justified or not, and whether the table are centered, figures are centered, captions are centered. Those are the few comments that we overlook "*

- Subject 7

There was an expressed frustration from authors when working with formatting in Word, as the software sometimes modifies the formatting in unintentional ways. Also, when converting to PDF the format may not look like it did in Word, leading to an extra check to make sure everything looks correct.

*"Word can really be a pain in the \*ss. You're trying to enter an extra line and it just makes maybe the previous line a heading or something."*

- Subject 8

Both authors and reviewers agree that there is a language barrier since many are not native English speakers. This leads to confusion about the report content as well as the feedback provided by the reviewers. Clarity, incorrect punctuation, typos and bad grammar are important problem areas in the reports. Numbers and units across the report may also not be consistent when delivered for review.

*"We are not English-speaking people, so that is one more thing that may block our thoughts or block the way we present in the reports."*

- Subject 6

**Initial poor quality of reports:** Most of the interviewees mention that the quality of the written reports is sometimes very poor. Particularly in cases of cultural clashes and where language barriers are particularly evident. If the report is bad from the beginning, it usually means that it will circulate in the process many times before it can finally be approved, which takes extra time and resources.

*"So, the main problem is that you get too bad reports from the beginning and then it kind of just loops around."*

- Subject 5

It is expressed that actually people who both write and review reports should undergo a shorter training beforehand in order to harmonize both quality, perspective, language and format. Today, this is completely lacking, as anyone can be selected as a reviewer and usually the person who conducted an analysis writes the report and this person may have no experience of writing actual reports.

*"I also think that the reviewers, it should be like a role that needs to be trained. Not everyone can do a good review."*

- Subject 5

**References:** The documents and files that are referred to in a report may sometimes not be properly released in the file management system. This means that the report itself cannot be released since its reference is not in a frozen state. This is an expressed issue by the reviewers which causes delay.

*"A report cannot be released if it has unreleased references. It can be a problem, especially with CAD-files"*

- Subject 3

Another issue is that the reports often come to the reviewers with information and statements lacking a proper reference.

*"Reference related mistakes happen quite often."*

- Subject 8

**Feedback and communication:** Some reviewers express that they do not like having to provide feedback and prefer to make the corrections themselves. It is expressed as hard to give feedback that reflects the reviewer's true intent.

*"Sometimes I feel like. It is hard to communicate what I want. To be changed in this report. Then like editing it myself. "*

- Subject 8

The authors may not understand it therefore. It can also be hard to understand due to the language barrier. Because of this, the mode of communication of feedback is sometimes very different as it might require an in-person meeting to clarify any doubts. It can also lead to the next revised report submission still containing the same or new problems since the feedback was not understood correctly.

*"Sometimes it can be really really hard to make the authors do what i want"*

- Subject 3

Also, communication in general between roles was expressed as problematic since there are many means to communicate. Some may prefer a standardized channel of communication. Another issue expressed by reviewers is that they feel like some authors do not learn from earlier feedback, leading to the same type of errors appearing repeatedly.

*"So, in some people, I don't see that they learn from the comments"*

- Subject 5

**Different opinions and subjectivity:** Writing the reports is a very person-dependent process, but even the final report can look very different according to the interviewees. However, this also applies to the reviewer, some have knowledge in the specific area, some do not, some are picky, some less so. This means that the process goes from being standardized where everyone follows similar templates to working methods and results spreading in different directions, which in turn can make the process take extra long because it clashes with individuals' opinions and experience. There is also a more noticeable difference between the working methods and knowledge of the different cultures in Europe and India.

*"It is actually an issue that there are so many ways to communicate and that everyone works differently"*

- Subject 6

#### 3.1.1.4 Not process specific issues

This theme represents different sub-themes that are not directly related to the analysis report generation process but have to do with other difficulties such as scheduling and time pressure.

**Agenda alignment:** A big factor delaying the report is that the people who review and approve are usually very busy.

*"Usually the time of matching everyone's time is a bottleneck. Author, reviewer, approver. The more you go down the chain, the less time they have."*

- Subject 8

When the task arrives at their desk, it takes considerable time for them to find time to start working. The challenge is that the agendas of the different people in the process are not aligned because they are busy or in different time zones. This greatly increases lead time, especially if the first report issue is very bad.

**Time pressure and human error:** When under time pressure humans tend to perform worse. This is an expressed issue both for authors and reviewers. Because

of other work responsibilities, you might also have to leave the report for a short while and then get back to it, continuing where you left off. This naturally leads to overlooking and missing certain things. Because of the high time pressure, reviewers are not often aware of the details of the original analysis that goes into to report, which will affect how they review it later.

*"We spend substantial time for the analysis, but when it comes to reporting, we get minimum time and try to explain as many things as possible, and I mean, being a human, you are bound to make mistakes."*

- Subject 7

Usually, the authors spend most of their time conducting the analysis. After this is finished, there may not be a lot of time left for them before the deadline to document everything in a very long report with the quality that is expected. This causes human errors such as inconsistent fonts and size and not centring tables properly.

## 3.2 Literature findings

This section presents the results of the literature research where each paper are summarized and listed including the title, authors and main findings. The literature review resulted in 11 papers that provided both inspiration for new concepts but also described the capabilities and limitations of LLMs. The references to all papers can be found at the end of each title.

1. **Title:** Chain of Thought Prompting Elicits Reasoning in Large Language Models [47]  
**Author:** Jason Wei, Xuezhi Wang, Dale Schuurmans, et al.  
**Publication date:** January, 2022.  
**Main findings:** This paper talks about whether Chain of Thought Prompting can increase the precision and credibility of the answers of an LLM. Chain of thought means that the LLM in question must think through the approach step by step. By dividing a task into smaller simple parts and solving them one by one, the model is more likely to get it right and arrive at an answer that has a higher probability of being correct.
2. **Title:** Communicative Agents for Software Development [38]  
**Author:** Chen Qian, Xin Cong, Wei Liu et al.  
**Publication date:** December, 2023.  
**Main findings:** This paper describes how ai agents run by LLMs with different roles and skills collaborate through a structured flowchart to simulate a real life office solving software development tasks. The agents benefit from functionalities such as role assignment, memory stream and self-reflection to solve the tasks assigned to each one. The solution is based on two agents interacting with each other loop by loop, until both are satisfied to pass on the result to the next pair of agents who continue in the same second. When the whole flowchart of agents is satisfied, the final result is given. The results

of the study indicate that the solution can help engineers with their tasks but even if the results are promising, challenges are mentioned with, among others, inconsistent outputs, difficulties with image modalities and when the model gets stuck in a loop without success in solving the given task.

3. **Title:** GPT4 is Slightly Helpful for Peer-Review Assistance: A Pilot Study [48]

**Author:** Zachary Robertson

**Publication date:** June, 2023.

**Main findings:** This paper discusses how GPT-4 can be used to perform reviewing tasks of scientific reports. The paper further addressed that the peer-review performed by GPT-4 was also done by a human afterwards where the performance of both was measured and compared. The results showed that the two reviews were equivalent on an overall level. However, it was mentioned that it would be interesting to further research the performance of reviewing low-level details in the reports. Some discussed conclusions were the difficulty for the model to give recommendations where e.g. judgment, empathy or experience would be necessary.

4. **Title:** Improving Factuality and Reasoning in Language Models through Multi-agent Debate [49]

**Author:** Yilun Du, Shuang Li, Antonio Torralba, et al.

**Publication date:** May, 2023.

**Main findings:** This paper first discusses how few-shot prompting has created promising capabilities for LLMs. The paper further discusses how the accuracy and the performance can be further increased by having several models either of the same kind or different models perform a task using the few-shot prompting methodology. Then, additional LLMs are allowed to answer the question but with access to the answers of the other LLMs. Thus, the next LLM in line gets examples of the output and can then reason and make decisions with more data. The results showed promising performance where for each iteration of "debate", the accuracy increased. However, the paper discusses limitations such as that the methodology being computationally expensive or that models sometimes have difficulty capturing the full context of the outputs from previous LLMs.

5. **Title:** Multi-Agent Collaboration: Harnessing the Power of Intelligent LLM Agents [50]

**Author:** Yashar Talebirad, Amirhossein Nadiri

**Publication date:** June, 2023.

**Main findings:** The article discusses, among other things, dividing complex tasks into smaller parts that are then distributed to different AI agents with different roles and specialties who then collaborate to solve the tasks. The results are described as promising with opportunities to apply the method in subject areas such as law to simulate, for example, a court room. Furthermore, challenges and limitations are discussed such as the problem of hallucinations and scalability as only a relatively small user case constitutes a complex sys-

tem.

6. **Title:** Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4 [51]  
**Author:** Sondos Mahmoud Bsharat, Aidar Myrzakhan, Zhiqiang Shen, et al.  
**Publication date:** January, 2024.  
**Main findings:** This paper talks about how accuracy and performance of an LLM can be increased just by prompt-engineering. That is, in a structured and “correct” way, give an LLM instructions on how to behave. They talk about 26 different guidelines that are tested and show a promising result where, among these points: "Give the LLM a role", "divide complex tasks into smaller easier instructions" or to simply say “Your task is:” are included.
7. **Title:** ReAct: Synergizing Reasoning and Acting in Language Models [52]  
**Author:** Shunyu Yao, Jeffrey Zhao, Dian Yu, et al.  
**Publication date:** October, 2022.  
**Main findings:** The report discusses the ReAct framework, which is based on a combination of the Acting and Reasoning methodology and LLM’s Chain of Thought. The methodology is based on first reasoning about what and how the “first” action should be taken. The model then observes the result and if it is not satisfied, it repeats the process by, for example, changing the action, changing the input or the approach in general until it is satisfied with the final result and responds. Since everything happens in the same prompt there is no need for a memory buffer. The results indicated an efficiency with high human interpretability and trustworthiness. The degree of hallucinations was also reduced. Limitations of the methodology are described as, for example, that more difficult tasks that need to be iterated several times quickly build up the limited context window.
8. **Title:** Reflexion: Language Agents with Verbal Reinforcement Learning [53]  
**Author:** Noah Shinn, Federico Cassano, Edward Berman, et al.  
**Publication date:** October, 2023.  
**Main findings:** Reflexion is a framework used by LLMs that according to the paper should learn from their mistakes to increase performance. This framework is described as being developed as a more flexible alternative to fine-tuning or updating the weights in the LLM. Reflexion uses tools such as: Actor - performs tasks, Evaluator - rates performance and results, Self-Reflection - reflects and gives recommendations, Memory - a buffer of saved performed, rated and reflected on tasks. Saved tasks are then included for future performed tasks and evaluations with the hope that the model "learns from its mistakes". The model showed good results where it is described as outperforming other decision-making approaches. A limitation is that it sometimes ended up in a situation where it chose a non-optimal choice of action.

- 
9. **Title:** Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? [54]  
**Author:** Sewon Min, Xinxi Lyu, Ari Holtzman, et al.  
**Publication date:** October, 2022.  
**Main findings:** This paper mainly discusses the importance of instructions in the input prompt for LLMs. Instead of zero-shot prompting, having only a few labeled demonstrations as input greatly increases performance. The model then has a better understanding of the problem at hand along with what it is expected to do and mostly does exactly as the instruction says. Despite the conflicting hypothesis, the results showed that the model understood even if random labels were assigned to the demonstrations. However, without both labels and demonstrations at all, performance dropped significantly.
  10. **Title:** Retrieval-Augmented Generation for Large Language Models: A Survey [55]  
**Author:** Yunfan Gao, Yun Xiong, Xinyu Gao, et al.  
**Publication date:** January, 2024.  
**Main findings:** This paper starts by discussing the problem of LLMs hallucinating or being trained on outdated data and thus the need for a solution to increase both accuracy and credibility. Retrieval-Augmented Generation (RAG) is a method based on equipping the LLM with a tool to be able to retrieve information from its own library of information. The library is in the form of a vector database. The methodology is then based on retrieving and utilizing the library of information to increase the precision of the answers. The paper further discusses that this means that an LLM can quickly become a specialist in a chosen field just by building the library with the necessary data. Despite the promising results, future challenges are discussed such as the difficulty of handling modalities other than text and the limited size of the context window.
  11. **Title:** The Rise and Potential of Large Language Model Based Agents: A Survey [56]  
**Author:** Zhiheng Xi, Wenxiang Chen, Xin Guo, et al.  
**Publication date:** September, 2023.  
**Main findings:** Everything from an AI agent’s origin, to construction, to functionality, to capabilities and disadvantages is discussed in this paper. The paper starts by explaining why an LLM is suitable as a basic component of an agent. The paper then goes on to discuss that an agent can be augmented with functionalities such as memory, knowledge bank, information retrieval, reasoning and reflection functionalities. In addition, the agent is given access to a set of tools to use to perform specific tasks. The basic idea is to let the agent, like a human, have a “brain” that is built on an LLM and then equip the agent with the above. The agent is then free to interact without supervision, think, reflect, use their toolbox and then act based on what the agent considers reasonable. Finally, the paper discusses agents’ capabilities such as, for example, task completion, sociability and the ability to develop and learn. However, with capabilities come risks such as trustworthiness and

### 3. Results

---

robustness in a broad set of tasks, among others.

# 4

## Concept development and testing

The overall methodology of the software development process was described in the methodology. This section describes each loop and concept development step of the agile methodology process in detail. To easily manipulate, store, share, monitor and evaluate code and results during the process, Github was used, see link [https://github.com/DaniSode/Master\\_Thesis](https://github.com/DaniSode/Master_Thesis). First, the design criteria to evaluate are presented. Then, performance measures are defined according to a set of definitions. Lastly, the developed concepts with respective program logic, prompts and generated outputs are presented along with justifications behind decisions made. At the end of each concept description, results are presented using the performance measures defined. The chapter concludes with a full performance comparison of all development efforts.

### 4.1 Design criteria to evaluate

The process of defining functionality to evaluate as described in section 2.3.1 resulted in the main goal of developing an LLM-based report reviewer, covering several aspects of the analysis reports. The intention was for the authors to use this tool to receive some feedback and make final corrections before delivering it to the real reviewer, reducing the risk of poor initial report quality. In addition, a function to efficiently retrieve and ask questions about old reports was chosen for the list. The full list of defined design criteria (DC) to evaluate is presented below in table 4.1.

ID	Design criterion	Description
1	Review language	Check the report for language errors as per the specific writing guidelines
2	Review report content	Check the report for incorrectly placed or missing content as per the guidelines
3	Review reference usage	Check if the references used in the report are correctly used and exist in the file folder
4	Review report consistency	Check if the report is consistent in its statements, methods used and results presented.
5	Review missing references	Check the report for statements missing a reference
6	Retrieve old reports	Retrieve old report content with a user query in natural language
7	Generate new parts of report	Retrieval of old report to use as inspiration and with a user query, generating parts of a new report.
8	Communicate to user	Present the functionality to the engineers

**Table 4.1:** The defined design criteria to evaluate.

## 4.2 Measuring performance

To measure how well the design criteria are fulfilled, performance measures were created as mentioned in 2.3.2. These are presented below.

### 4.2.1 Criteria 1-5

To evaluate performance on the report review criteria (DC1-DC5), a test case report was created for each. Each case is a modification of the original provided dummy report but injected with a set of intentional DC-specific errors based on what the real reviewers can expect to find in a newly authored report. An intentional error injected into a test case will be referred to as  $E_{intentional}$ .

All concepts with associated test cases were constructed to output the recommendations it found when assessing the full report to a text file that could be easily read and interpreted by a human being. Since an LLM does not produce identical outputs each time it is prompted, there were differences in the output between the test runs. Because of this, several test runs were made to evaluate performance on the design criteria.

It is important to note that there were many errors in terms of language, tense and references in the given original dummy report and the most obvious ones were corrected manually before creating test cases. However, this meant that most models with their associated tests gave good and valuable feedback but on errors that were not in the set of  $E_{intentional}$ .

To specify how performance was measured on DC1-DC5, the following three definitions were used:

- Spotted intentional error  $F_{intentional}$ : A point of feedback that can be used to easily track down an intentional error in the test case report without any further inference or analysis.
- Not accurate error  $F_{false}$ : A point of feedback that is untruthful, unhelpful and does not provide any value to the engineer
- Additional useful feedback  $F_{add}$ : Points of feedback that are truthful and helpful but not related to the set of intentional errors. (NOT measured due to the fact that the given dummy report was previously approved by an analysis lead and considered correct, this would also be VERY time-consuming)

The classification according to the definitions above of the generated feedback in the different test cases was assessed by human evaluation. The definitions were then used to define two general performance measures, as shown below in equation 4.1 and 4.2:

The proportion of intentional errors spotted by the concept:

$$P_{intentional} = \frac{\sum F_{intentional}}{\sum E_{intentional}} \quad (4.1)$$

The proportion of accurate feedback:

$$P_{accurate} = \frac{\sum F_{intentional}}{\sum F_{intentional} + \sum F_{false}} \quad (4.2)$$

In addition to these defined measures, token-usage and execution time were also measured for each concept. It is important to emphasize that because of different concepts and test cases, the type of output was also different. The quality or correctness of the feedback may therefore be more or less subjective, and therefore harder to assess.

### 4.2.2 Criteria 6-7

Criteria 6 and 7 take a different turn in terms of the functionality of the concepts to focus on retrieval of report content or the generation of new output rather than commenting on previously written text. Due to the lack of writing and reviewing experience within the engineering domain, but also the lack of time and engineers with the right experience a proper test case to evaluate performance could not be made. Therefore, the criteria with their corresponding concepts were rather assessed as a proof of concept and no real evaluation of the criteria was made.

### 4.2.3 Criteria 8

The final design criteria is about presenting the functionality to the engineers and allowing them to interact with it through a real user interface. Since the main

scope of the project was about the capabilities and limitations about the LLMs themselves, the solution to this criteria was intended as "proof of concept" rather than fully tested and evaluated. Because of this, no real test of it was made.

### 4.3 Concepts

With a defined framework for measuring performance, the development of concepts to fulfil DC1-8 started. All of the developed concepts made use of the base-models Mistral 8x7B and OpenAI GPT-4, to explore and compare an easily accessible open-source model to the current state-of-the-art. The concepts created were all based on 100% Python and the Langchain framework. It should be stated that it took a significant amount of time to learn about the framework to be able to properly implement the solutions. Because of this, there was a lot of trial and error and smaller sub-concepts that are not mentioned in this report. The following sections will describe the most important aspects of each approach. As mentioned, the full implementation in terms of code, prompts, outputs and documents used can be found on the thesis Github.

#### 4.3.1 Parsing the analysis reports

The analysis reports are created by the engineers using Microsoft Word. To efficiently perform operations on the content of the analysis reports, a word parser script *read\_docx\_report.py* was developed to store the report content. The script was made based on the word template used by the engineers, and therefore expect a certain format of the headings. The content of the report is stored in a Python dictionary with each heading as a key and the corresponding section text as the value, both in a string format. The script also includes all tables in the report and parses them into a CSV format with a "|" as the delimiter. This format was proven among the most effective for LLM understanding as stated in [57]. The report parser was a crucial component in all succeeding developments.

#### 4.3.2 Design criteria: Review language

The evaluation of the first design criteria of reviewing the report language according to the specific report writing guidelines resulted in 3 different approaches to the problem. The test case created contained 11 intentional errors that explicitly broke the specific writing guidelines provided as presented in appendix B.2. Several types of errors were injected across the report such as mentioning of the customer's name, personal names, wrong verb tense and inclusion of inappropriate personal pronouns. The full test case, containing a description of each error and its location in the report is presented in appendix C.1. The full results of all official test runs for each of the developed concepts are also found in appendix C.2.

### 4.3.2.1 Language 1

The first concept, referred to as Language 1 was a very simple approach because of the desire to start small and quickly get the ball rolling to evaluate performance. The strategy was to evaluate the full report in one go to keep token usage at a minimum and focused on getting all the basic functionality in place such as the report parser and correct invoking of and switching between base-models.

As stated in [54] LLMs have the ability to in-context learn, meaning that they based on some input label pairs can perform a new task by inference alone with impressive performance. These input label pairs can be seen as demonstrations of the desired output of the model. This knowledge gave inspiration for the core idea of Language 1, which is to provide the model with the report-specific writing guidelines and some examples of good and bad sentences. This was done through the definition of a system prompt, which first gives the model some additional personality before passing on the full writing guidelines along with good and bad sentences under the key `{writing_guidelines}`. This way of assigning a role to an LLM is a commonly used prompt principle to achieve better answers [51]. This system prompt used for Language 1 is shown below in prompt 4.1:

```
You are a helpful reviewer of engineering analysis reports within
the aircraft industry. Your only responsibility is to check that
the language in the report aligns with the writing guidelines.
The guidelines for writing the report are as follows:

{writing_guidelines}
```

**Prompt 4.1:** System prompt used in Language 1.

Furthermore, the prompt used to generate output from the model was designed to structure the feedback to several JSON objects, each containing the error type, the incorrect sentence and a suggestion for a corrected sentence. This decision was taken after several rounds of prompt refinement to be able to evaluate the output more easily. Lastly, to allow the model to assess the report text, the analysis report was parsed and stored in a string. The parsing was done excluding the sections summary, revision record, table of contents and references because they did not include a lot of text. The string created was then fed into the prompt described under the key `{paragraph}`, see prompt 4.2.

After running this concept on the test case defined with Mistral 8x7B as the base model, it was quickly determined that it did not produce any  $F_{intentional}$ . Rather, it produced a hallucination-mess characterized by an obsession with the present simple verb tense. In addition, it gave feedback on parts of the input label pairs inside the system prompt. Several examples of this undesired output ( $F_{false}$ ) are presented in output 4.1. On the other hand, when running the concept with the current state-of-the-art GPT-4 as the base-model, several  $F_{intentional}$  was generated. Examples of these are shown in output 4.2:

#### 4. Concept development and testing

---

Please check the following part of the report:

```
{paragraph}
```

Your feedback must include the following:

Error type: The type of error as specified in the guidelines,  
Incorrect sentence: the sentence containing the error itself,  
Corrected sentence: Correct the error and provide the corrected version of the sentence

Answer with each point of feedback as a JSON object with the following keys:

```
"Error type":  
"Incorrect sentence":  
"Corrected sentence":
```

Be thorough and think about what you are doing.

#### Prompt 4.2: Prompt used to generate output in Language 1

```
{  
  "Error type": "Present simple verb tense",  
  "Incorrect sentence": "The Limit analysis for the 360-degree solid model has been performed according to DP [4] and DVP [3].",  
  "Corrected sentence": "The Limit analysis for the 360-degree solid model performs according to DP [4] and DVP [3]. (changed to present simple verb tense)"  
},  
{  
  "Error type": "Present simple verb tense",  
  "Incorrect sentence": "This report goes more in detail with investigations than in previous analyses [13][15] which are not stated in DVP [3] and DP [4].",  
  "Corrected sentence": "This report goes more in detail with investigations than in previous analyses [13][15] which not state in DVP [3] and DP [4]. (changed to present simple verb tense)"  
},  
{  
  "Error type": "Present simple verb tense",  
  "Incorrect sentence": "Linear static analysis of the 360 model has been carried out with 115% mechanical (limit) loads and redline thermals [11]",  
  "Corrected sentence": "Linear static analysis of the 360 model carries out with 115% mechanical (limit) loads and redline thermals [11]. (changed to present simple verb tense)"  
},  
}
```

**Output 4.1:** A truncated output from running Language 1 with Mistral 8x7B, showing three  $F_{false}$ .

```

{"Error type": "Use of personal names",
 "Incorrect sentence": "The estimated plastic strain from the
 interpolation of stress strain at 900K for Material A for the
 obtained von Mises stress of 601 MPa (see Figure 9-5) is found by
 Alex to be 0.21%",
 "Corrected sentence": "The estimated plastic strain from the
 interpolation of stress strain at 900K for Material A for the
 obtained von Mises stress of 601 MPa (see Figure 9-5) is found to
 be 0.21%"},

{"Error type": "Use of first person",
 "Incorrect sentence": "I have been carrying out a non-linear
 analysis for the DLCs which has been resulting in negative linear
 analysis margins (i.e. where some plasticity is found).",
 "Corrected sentence": "A non-linear analysis for the DLCs which has
 been resulting in negative linear analysis margins (i.e. where
 some plasticity is found) has been carried out."},

```

**Output 4.2:** A truncated output from running Language 1 with GPT-4, showing two  $F_{intentional}$ .

The official testing of the concept consisted of 2 runs of Mistral and 1 run of GPT-4. The average performance on  $P_{intentional}$  and  $P_{accurate}$  for the GPT-4 run was 45% and 63% respectively. Mistral did not produce any significant results.

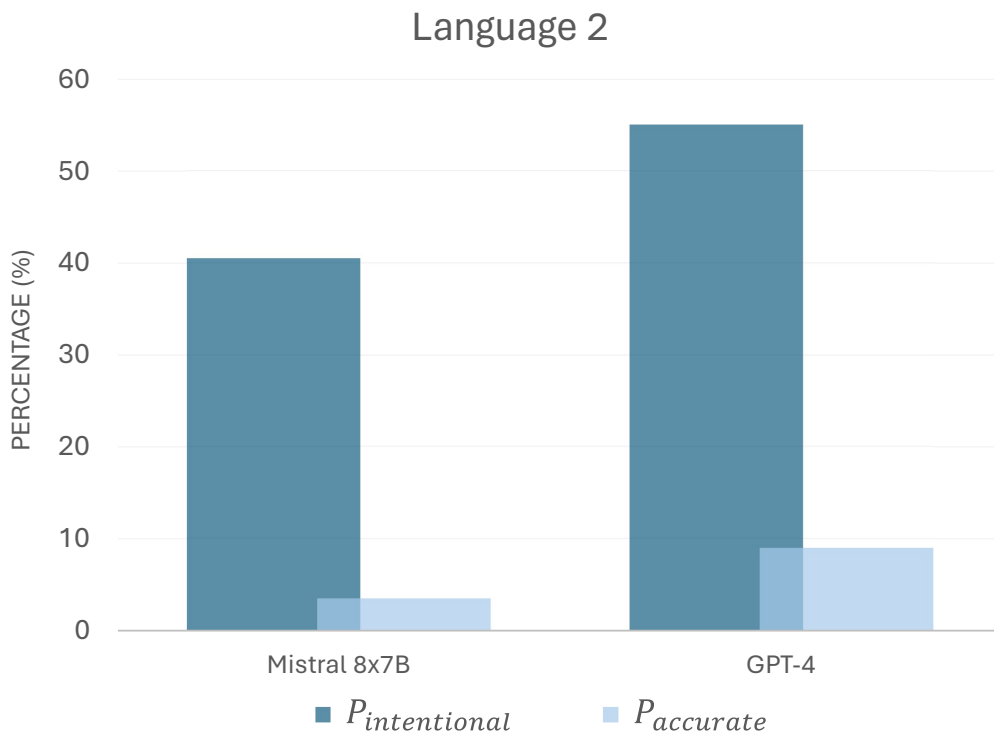
#### 4.3.2.2 Language 2

With Language 1 created and tested, a new concept with the goal of increasing the hitrate of spotted intentional errors was created and given the name Language 2. Since LLMs may struggle when prompted with a lot of information at the same time [51], the idea with Language 2 was to divide the full report into smaller chunks to explore whether this would increase  $P_{intentional}$  and  $P_{accurate}$  at the expense of higher token usage, since each chunk would be prompted together with the full guidelines.

The logic of the program and prompts used are similar to Language 1. The main difference is that after the report parsing step, the full report string was recursively split with a size of 500 characters and with the separators ["\n\n", "\n", ". "] to ensure that no sentences were being cut halfway through. This split size was chosen after some iterations of trial and error. Prompts are also slightly modified to accommodate that the model is being fed with a chunk of a report instead of the full. Also, the model is prompted to answer with <NO\_FEEDBACK> if it does not think there are any errors in the report chunk. Lastly, in an attempt to remove any feedback given on parts of the prompts that are not part of the report, a sentence **Only give feedback on the section given!** was added. In this concept, each chunk of the report was fed to the prompt under the key {paragraph} using the batch command in Langchain, resulting in 55 separate runs in parallel to assess the whole report.

After starting the testing it was seen that Mistral started to produce some  $F_{intentional}$ ,

which was not the case in the previous concept. GPT-4 also increased  $P_{intentional}$  compared to the previous approach. However,  $P_{accurate}$  dropped significantly in the GPT-4 run due to a large increase in output of  $F_{false}$ . It was also very low for the Mistral runs. This is not ideal performance since only 1 in 10 points of feedback is of use. Token usage is also much greater in this approach. The official testing of Language 2 consisted of 2 runs of Mistral and 1 run of GPT-4. The average performance on  $P_{intentional}$  and  $P_{accurate}$  for both base models are presented below in figure 4.1



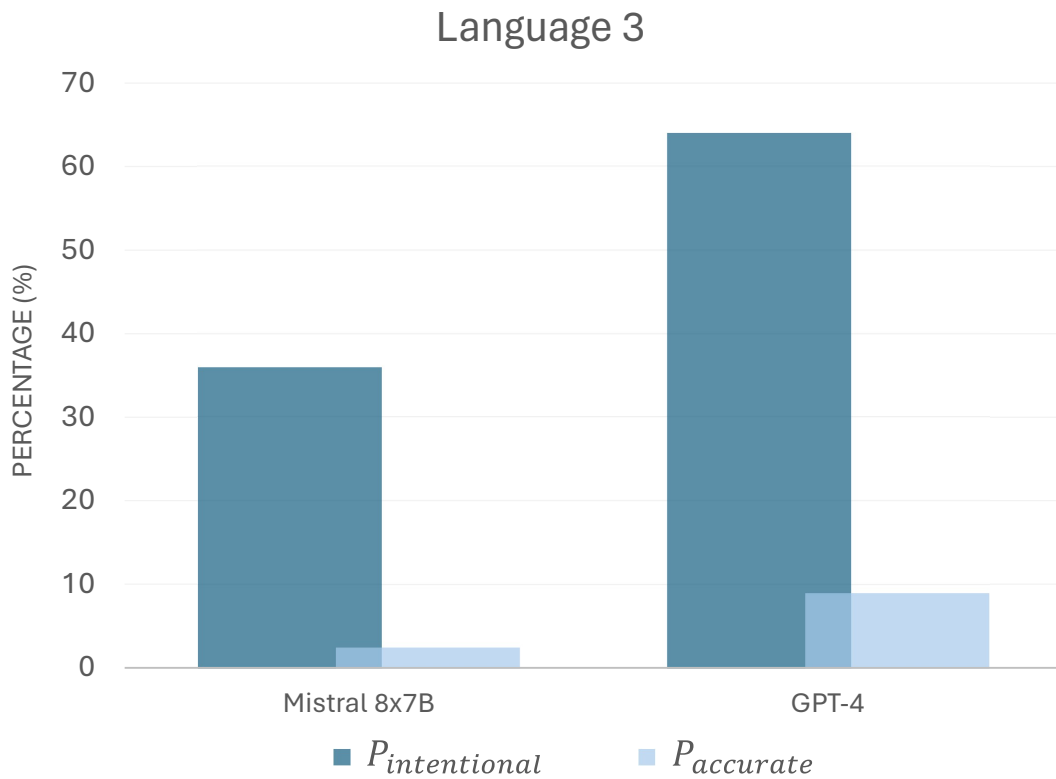
**Figure 4.1:** Average results of Language 2

#### 4.3.2.3 Language 3

In a final attempt to boost performance, the concept Language 3 was created. The goal was to explore if a further reduction in prompt size would have any significant effect on the output.

The approach takes inspiration from [50], in which the authors discuss collaboration between several LLMs to achieve a goal, each having its own speciality. The idea in Language 3 is to divide the language guidelines given into 3 parts, assigning each to a separate LLM. This was done by using the prompt 4.1 but with one third of the writing guidelines as well as the input label pairs as a system prompt for 3 different LLMs. To review the full report, each chunk of report is assessed by each LLM and their respective guideline specialties, resulting in 3 calls for each chunk. This means that a lot of calls need to be done to get through the report, which is computationally expensive.

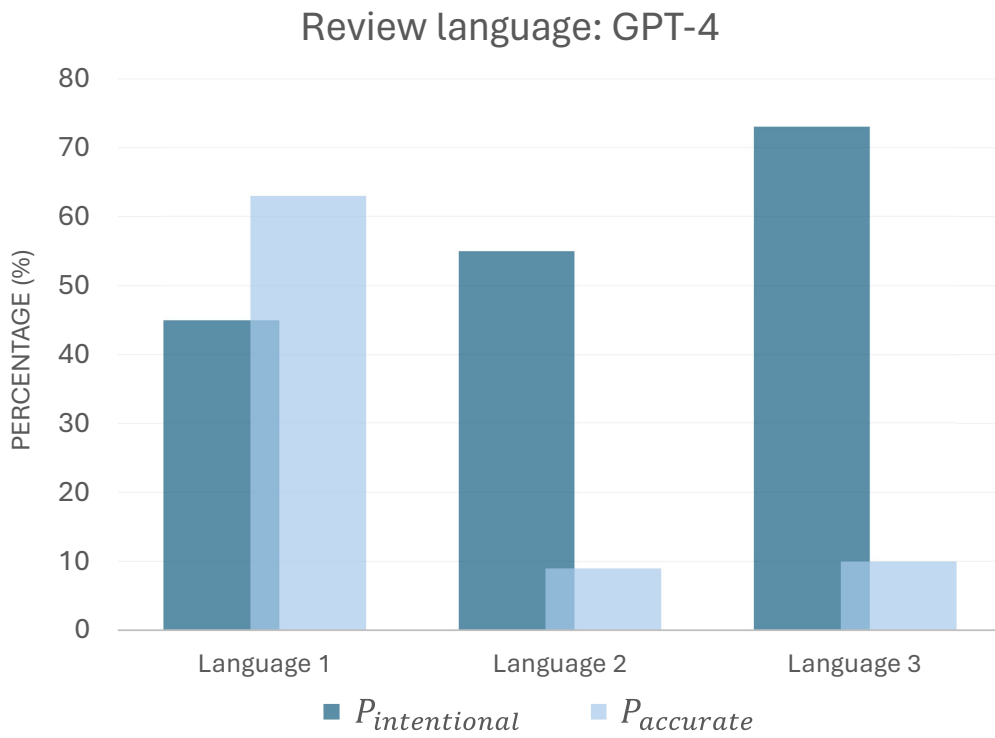
Running the concept shows a further increase in  $P_{intentional}$  for the GPT-4 run. Mistral continues to struggle greatly and shows a decrease in performance both on  $P_{intentional}$  and  $P_{accurate}$ . The official testing of Language 3 consisted of 2 runs of Mistral and 1 run of GPT-4. The average performance on  $P_{intentional}$  and  $P_{accurate}$  for both base models are presented below in figure 4.2



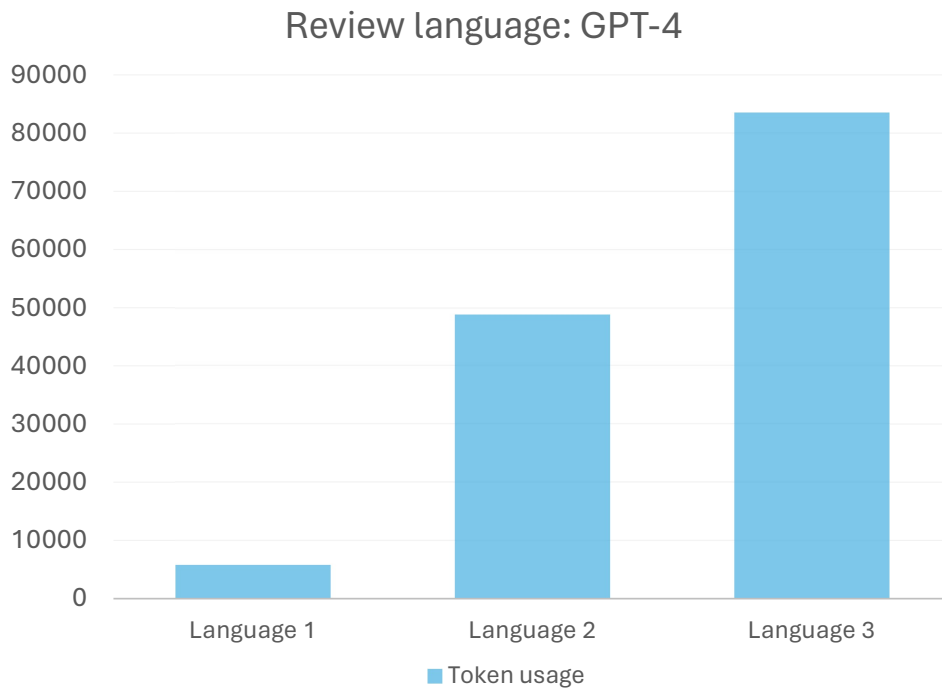
**Figure 4.2:** Average results of Language 3

#### 4.3.2.4 Comparison of approaches

When running on GPT4, the three concepts show an increase in  $P_{intentional}$  but a decrease in  $P_{accurate}$ , see 4.3. Token usage also increases greatly for each new concept, see 4.4.



**Figure 4.3:** Results on DC1 using GPT-4



**Figure 4.4:** Token usage on DC1 using GPT-4

### 4.3.3 Design criteria: Review content

The evaluation of the second design criteria of reviewing the report content according to the specific report content guidelines resulted in 2 concepts. The test case created contained 15 errors that broke the specific content guidelines provided as presented in appendix B.3. The types of errors injected across the report were of different kinds such as the purpose of the analysis missing from the introduction, an added non related section about playing innebandy on fridays and several mentions about the methodology in the results section. The full test case, containing a description of each error and its location in the report is presented in appendix C.3. The corresponding results of all official test runs for each of the developed concepts are found in appendix C.4.

#### 4.3.3.1 Content 1

With the learnings gained from DC1, the first concept to check the content in the report was based on assessing it section by section. A system prompt is defined to give the model the desired personality. A version of this prompt was created for each section of the report by feeding it with the respective content guidelines under the key `{content_guidelines}`. This prompt is shown in prompt 4.3. Output is then generated with prompt 4.4, taking an entire section of the report such as the introduction under the key `{paragraph}`.

```
You are a helpful reviewer of aerospace engineering reports. Your
only responsibility is to check that the section you are given
contains the content specified in the following guidelines:
```

```
{content_guidelines}
```

```
No other types of content is allowed.
```

```
Answer with a concise enumerated list containing each error along
with a description. If there are no errors, answer with only <
NO_ERRORS>.
```

**Prompt 4.3:** System prompt used in Content 1.

```
Please review this report section: '{paragraph}'
```

**Prompt 4.4:** Generation prompt used in Content 1.

Initial testing showed that both Mistral and GPT4 could pick up quite a few intentional errors and produced significantly less bad feedback as opposed to the DC1 concepts. To provide some examples of Mistral's  $F_{intentional}$  when assessing the requirements section, see output 4.3. Output 4.4 shows a  $F_{intentional}$  from mistral when assessing the results section. On the other hand, the tendency to produce false feedback remains. Output 4.5 shows some undesired feedback, also produced by mistral:

#### 4. Concept development and testing

---

```
2. The source of the requirement is not clearly stated for each
requirement in the table.
3. The requirement "Innebandy is played on Fridays not on any other
days, this is due to Alex Friday innebandy rule. If his minions
would play innebandy on any other day than Fridays then suitable
punishment will be applied." is not a relevant aerospace
engineering requirement and should not be included in this section.
```

**Output 4.3:** A truncated output from running Content 1 with Mistral, showing two  $F_{intentional}$  from the requirements section

```
- The section includes information about the process: "After
removing the duplicates, 23 unique limit load cases are identified
for limit analysis on the 360 FE model." This information should be
in the method section.
```

**Output 4.4:** A truncated output from running Content 1 with Mistral, showing a  $F_{intentional}$  about mentioning of the process in the results section

```
- The section contains a reference to the Finite Element (FE) model
report, which is the 'FE model details' part.
.....
1. The section does not explicitly refer to the Design Practice (DP
) and Design Verification Plan (DVP) that drives the analysis
method in this report, as stated in guideline 1.
```

**Output 4.5:** A truncated output from running Content 1 with Mistral, showing two  $F_{false}$  given in the Analysis models and Method section

The second point of feedback output in 4.5 is easily classified as  $F_{false}$  when reading the actual content of the method section in the report, as stated below:

"The Limit analysis for the 360-degree solid model has been performed according to DP [4] and DVP [3]. This report goes more in detail with investigations than in previous analyses [13][15] which are not stated in DVP [3] and DP [4]."

- Method section

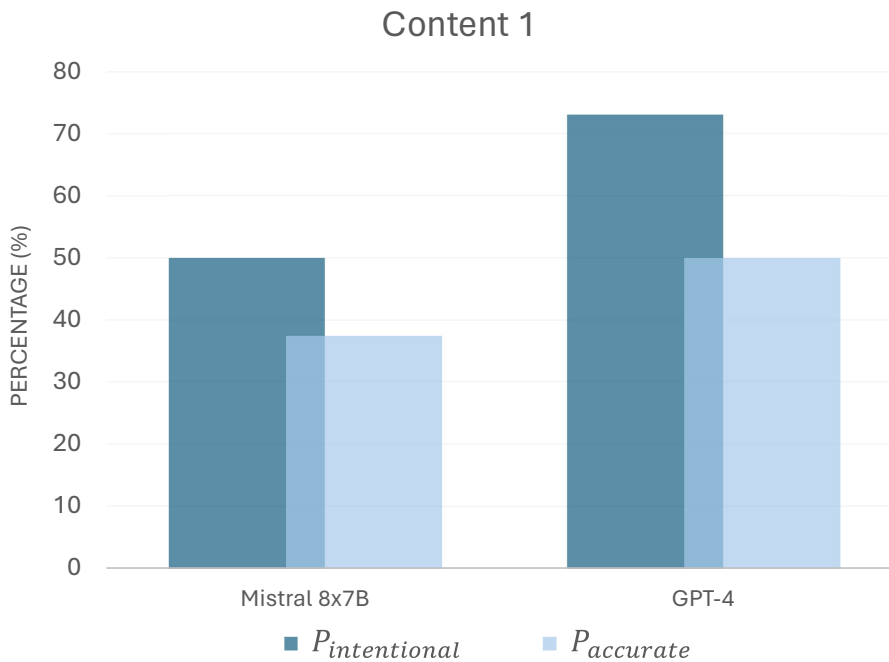
Similar to the results on DC1, GPT-4 shows significantly better performance compared to mistral, catching almost all intentional errors. It picks up the error regarding the missing reference to the Finite Element model report as seen in output 4.5 where mistral fails, see output 4.6.

When assessing the feedback provided by GPT4, it had a lot to say about the level of detail in the explanations in the report. This could be useful feedback, but not measured since it was not the explicit goal of the solution. It was also very tough on any type of content not mentioned in the guidelines. The official testing of Content

```
1. Error: The section does not contain a reference to the Finite Element model report.
```

**Output 4.6:** A truncated output from running Content 1 with GPT-4, spotting an error that Mistral did not.

1 consisted of 2 runs of Mistral and 1 run of GPT-4. The average performance on  $P_{intentional}$  and  $P_{accurate}$  for both base models are presented in figure 4.5.



**Figure 4.5:** Average results of Content 1.

#### 4.3.3.2 Content 2

The development so far had shown a large tendency to produce not useful or bad feedback. Just like a human reviewer, the feedback generated on the first try might be far from perfect if only given a short amount of time to think about what to point out. Usually, some time to reflect and think is required for a human to answer satisfactorily when dealing with more complex tasks. In [58], the authors take inspiration from this and introduce an approach referred to as *self-refine* to mimic the human behaviour of reflecting upon something that has been said or written. In the approach, an LLM generates an initial output, that output is then given critique by itself which is used to refine the initial output. This can be done in a loop several times until a stop condition is met. Across the evaluated tasks in the paper, outputs created through self-refine were preferred by human evaluators.

In an attempt to reduce  $F_{false}$  such as the ones shown in output 4.5, a concept taking inspiration from *self-refine* by the authors in [58] was developed and referred

to as Content 2. The approach contains 3 system prompts to create LLMs with different personalities, the first is similar to prompt 4.3 but with the content guidelines excluded. This is referred to as the generation prompt and was used to produce the initial output. In addition to this, a reflection prompt was defined and tuned to give feedback on whatever is produced by the generation prompt. The reflection prompt used is shown in prompt 4.5. Lastly, a refine prompt was defined with the purpose of implementing the feedback to create a refined output. See prompt 4.6.

```
You are a helpful reviewer of aerospace engineering reports. Your task is to check some errors found in a report section regarding its content. Do you agree with the errors spotted? Think about the guidelines and the report section provided, then mark each error with YES or NO depending on if you agree or not.
```

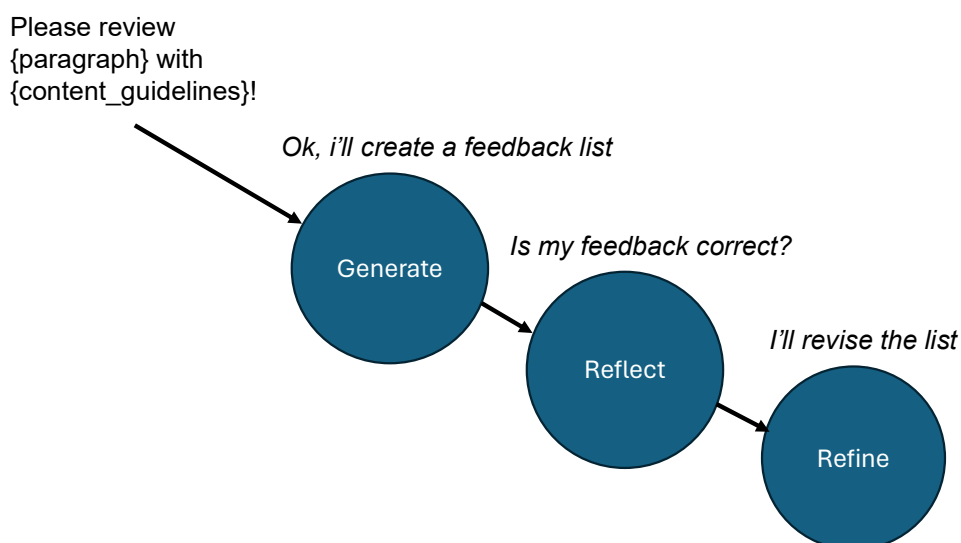
**Prompt 4.5:** The reflection system prompt used in Content 2.

```
Your task is to create an updated enumerated list with the mentioned errors, keep the ones marked with YES and discard the ones marked with NO.
```

```
Present the updated list clearly describing the remaining errors, make sure that any discarded errors are removed from the new list.
```

**Prompt 4.6:** The refine system prompt used in Content 2.

The user input in this case is a simple query asking for feedback on the report section {paragraph} with associated content guidelines {content\_guidelines}. A schematic of the idea as a whole is shown in figure 4.6



**Figure 4.6:** A schematic of the approach in Content 2.

As mentioned, this approach can be done in a loop to iteratively refine the output created. There was a development effort to let a separate LLM decide to stop the loop when all points of feedback were agreed upon. However, this approach proved to be very inconsistent between base-models and did not perform satisfactorily. therefore the decision was to continue with only one round of refinement.

Some interesting behaviour is observed when testing this approach. First of all, Mistral does not appear to be capable of performing the final refining step to get rid of thoughts it did not agree with at first. Because of this, Mistral was evaluated based on thoughts presented in the second reflect step instead. In the case of GPT-4, several incorrect points of feedback are spotted and removed from the final output as desired. See output 4.7 and 4.8.

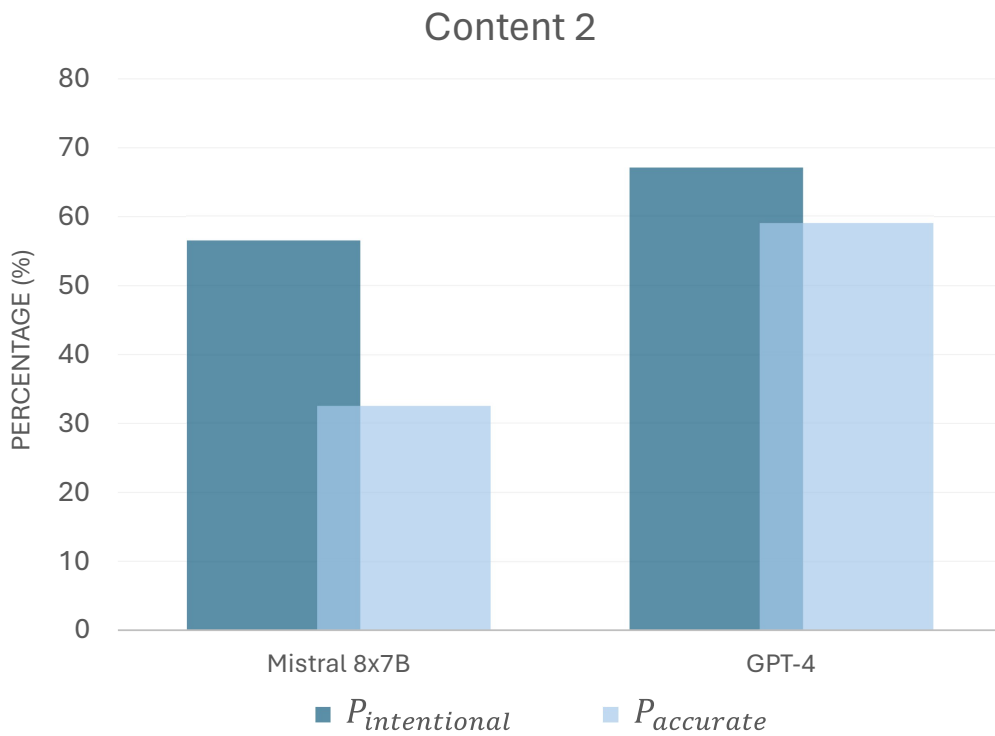
```
1. Error: The report section does not contain an explicit reference
to the Design Practice (DP) and Design Verification Plan (DVP)
that drives the analysis method in this report.
Description: The guidelines require an explicit reference to the DP
and DVP that guide the analysis method. This is missing in the
report section.
...
5. Error: The report section does not contain a subsection called '
Method Procedure'.
Description: The guidelines require a subsection called 'Method
Procedure' that details if the method used in the analysis deviates
from the DP or DVP and any additional calculations not specified
in the DP or DVP. This is missing in the report section.
```

**Output 4.7:** A truncated partial output from the initial generation step in Content 2 with GPT-4, giving two  $F_{false}$  when reviewing the Method section.

```
1. Error: NO
Description: The report does mention the Design Practice (DP) and
Design Verification Plan (DVP) multiple times throughout the text,
indicating that they guide the analysis method.
...
5. Error: NO
Description: The report does contain a subsection called 'Method
Procedure' that details the method used in the analysis, including
deviations from the DP or DVP and additional calculations not
specified in the DP or DVP.
```

**Output 4.8:** A truncated output from the reflect step in Content 2 with GPT-4, correctly realizing that the two points of feedback are incorrect.

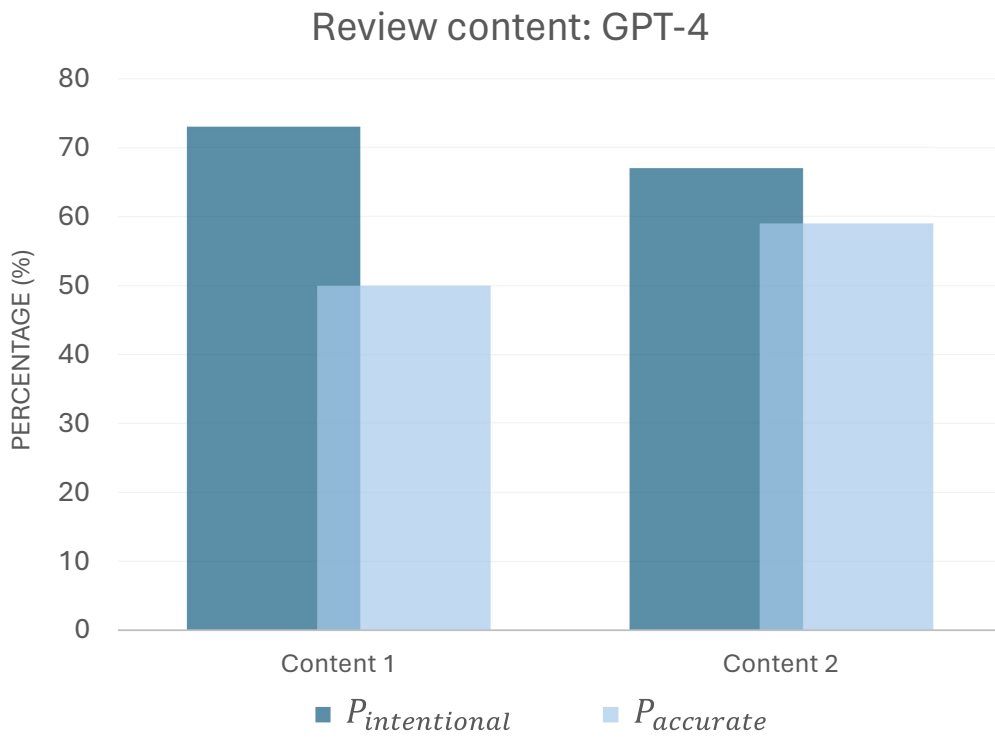
These are then neatly removed when refining to produce a higher  $P_{accurate}$  in the case of GPT-4. Mistral on the other hand did not show any improved performance using this approach. The official testing of Content 2 consisted of 2 runs of Mistral and 1 run of GPT-4. The average performance on  $P_{intentional}$  and  $P_{accurate}$  for both base models are presented below in figure 4.7.



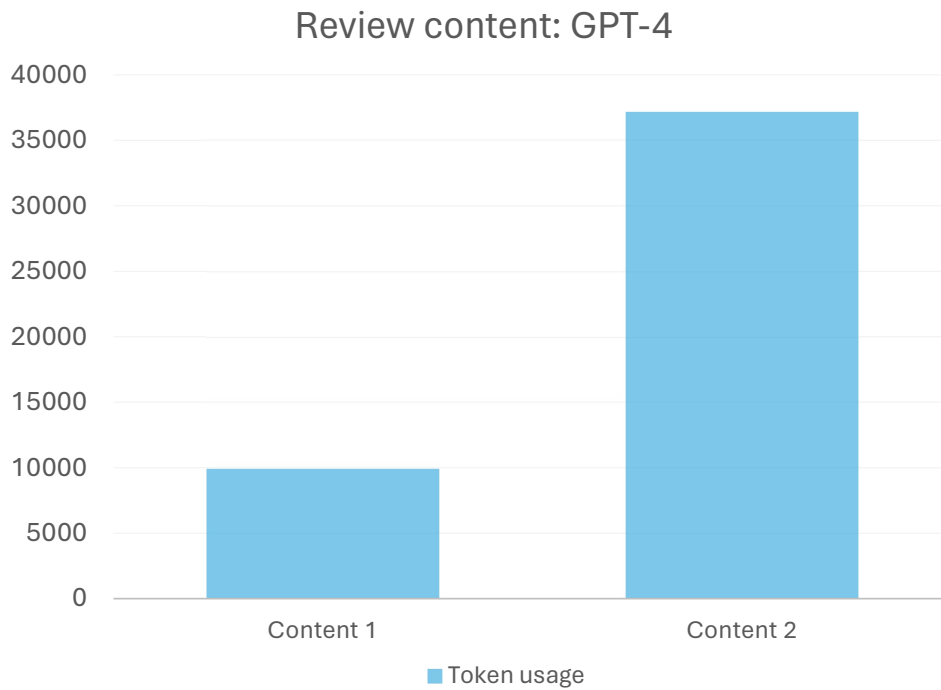
**Figure 4.7:** Results of testing Content 2.

#### 4.3.3.3 Comparison of approaches

The results show an increase in  $P_{accurate}$  after the implementation of the reflection step, see 4.8. The results of the tokens required to assess the full report for both content 1 and content 2 are presented below in 4.9. Due to the extra prompting in Content 2, token usage is significantly higher.



**Figure 4.8:** The final results on DC2 using GPT-4.



**Figure 4.9:** Token usage to assess the full report for both content concepts.

### 4.3.4 Design criteria: Review reference usage

The development of concepts to fulfil DC3 of reviewing the reference usage in the report resulted in three different approaches, two of which were officially tested. To be able to evaluate this functionality, 6 dummy references from the reference list of the report were created using ChatGPT. Each was around 1-2 pages long and had content that would simulate the real ones used at GKN Aerospace. The test case created for this DC consisted of 12 errors such as switching places of references in the text and adjusting the report to say something slightly different than what is stated in the reference. The full test case, containing a description of each error and its location in the report is presented in appendix C.5. The full results of all official test runs for each of the tested concepts are also found in appendix C.6.

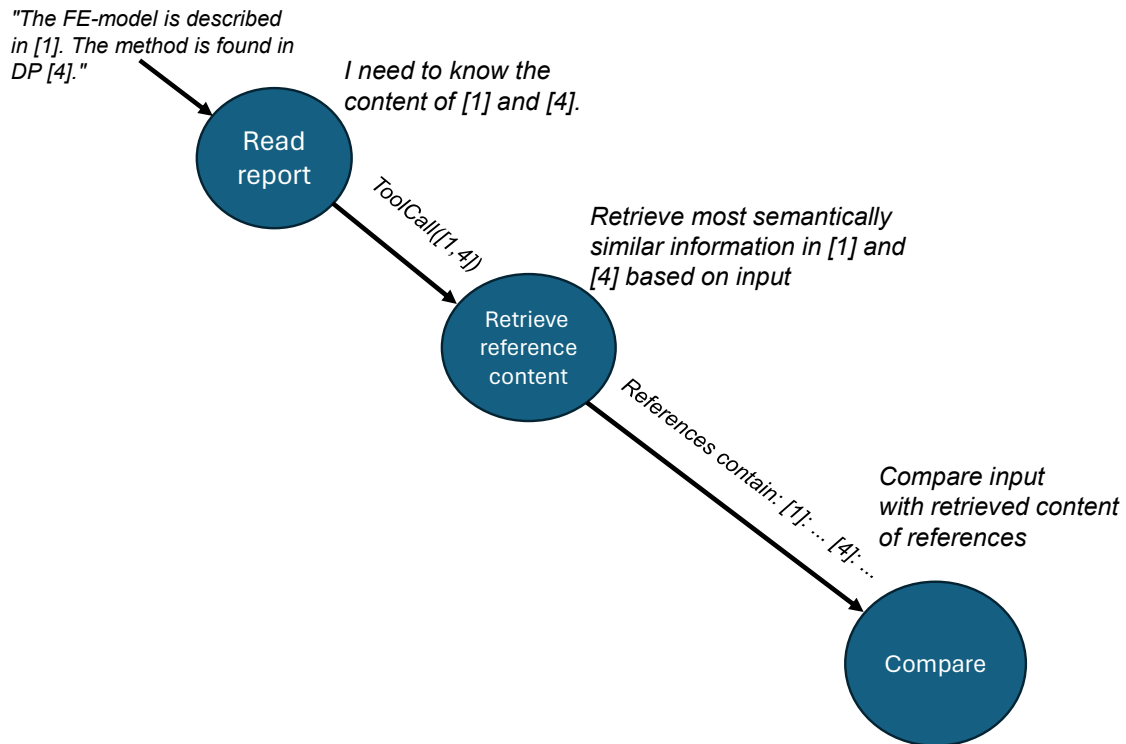
#### 4.3.4.1 Correct reference 1

The initial development efforts were about assessing if a larger prompt approach could still deliver any significant performance. This resulted in a concept which took in a chunk of report text along with the full content of the references used and queried about consistency between the report and used references. This worked quite well as long as there was a low number of references and not a lot of text in each. The knowledge from previous development that longer prompts appear to give worse  $P_{intentional}$  meant that new ideas had to be explored, such as efficient retrieval strategies to avoid long prompts.

Retrieval augmented generation (RAG) has proven to be a very efficient way to leverage Large Language Models when working with external databases that go beyond the model's training data. RAG is a flexible and scalable way to inject relevant chunks of data from large databases into a prompt to produce accurate answers [55]. In the case of developing solutions to DC3, this technique was of interest since a flexible mechanism to retrieve data from a large database of references was required to construct concise prompts.

Besides text generation, LLMs also possess the ability to use tools for different purposes. Just like a human reviewer can decide to go and read the references after seeing them mentioned in the report text, hypothetically, an LLM can decide to take a similar action and use a specific tool for it. As discussed in [56], access to tools can strengthen LLM's capabilities in complex tasks.

The development efforts on DC3 took inspiration from RAG and LLM tool-use to create the concept Correct reference 1. The system was designed using LangGraph from Langchain, which makes use of several nodes that communicate with different functionality. The simplified logic of this approach is shown in figure 4.10.



**Figure 4.10:** Simplified schematic of the approach in correct reference 1

More specifically, the program does the following: Read the reference list in the beginning of the report and store each filename in order inside a python list. This is to be able to index the list later on when numbered references occur in the report text. Then, loop over each dummy reference file in the specified folder, split the content of each into chunks and create embeddings which are stored in a vector database. The vector database is then used to define a Langchain retriever object which can be invoked to retrieve chunks of information that are semantically similar to the input query. This results in the definition of a retriever object for each reference in the folder. These are then stored in a dictionary under the key of their filename.

An LLM-invokable tool `get_reference_content(reference_numbers)` is defined. This tool takes the reference numbers seen in a chunk of report text and outputs a list of dictionaries created in the earlier step, each containing the reference number as the key and its corresponding retriever object as the value. In case the reference is not in the specified folder, a string "Not found!" is stored instead. In order for an LLM to use a tool effectively, the tool needs a description of how and why it can be used [56]. This description is shown below in the docstring of the function, see figure 4.11

```
"""
Finds the content of references given their numbers.

Args:
    reference_numbers: All the reference numbers spotted in the
                        report paragraph.
"""
```

**Figure 4.11:** The docstring of the reference retriever tool.

A tool calling agent is then created which has access to `get_reference_content`, this agent uses system prompt 4.7 for the desired personality. This agent makes up the node "Read report" in figure 4.10, and is responsible for taking action upon reading a full section of the report.

```
You are a helpful assistant, your responsibility is to find the
content of the references used in some report text. Only look for
references with this format: [x]. Do NOT consider figure, table or
section references.
```

**Prompt 4.7:** System prompt used for the first node in the system of Correct reference 1

Either, it intelligently produces a toolcall to the tool `get_reference_content` with the references seen or decides to move on to the next report section if there are none. Testing this on GPT-4 showed very good reliability in this aspect. Mistral was not capable of doing this reliably, which is why it was excluded from further development and testing. Below is an excerpt from the File Management section, and in 4.9 it can be seen how the agent in the first node decides that no action is needed.

"All files required to rerun of the analysis presented in this report are held in the AP1000 TRS Long-Term Storage under the registration number DB131156/01."

- File Management section

```
This paragraph does not contain any references in the specified
format: [x]. Therefore, there's no need to look for any reference
content.
```

**Output 4.9:** The first node in correct reference 1 deciding that it does not need to go look for reference content.

In case the first node decides it needs to fetch reference content, the next node in the graph is responsible for executing the tool call of `get_reference_content` with the list of spotted references as the input argument. The retriever objects that the

tool returns are then invoked one at a time using the report section of interest as the query. This outputs the chunks of reference that are most semantically similar to the report section. In case there is no retriever object for the reference spotted, "Not found!" is output instead, meaning that the solution as a whole can also pinpoint which references do not exist in the specified reference folder. An example of the output from retrieval of reference content after reading the conclusion section is seen below in output 4.10:

```
[4]:
Perform sensitivity analyses and parametric studies to assess the
influence of various factors, such as material properties,
geometric configurations, and operating conditions, on the response
of the component to different load cases"
...
[8]: Not found!

[2]:
Analytical reports detailing the assessment of limit load capacity
and permanent deformation for the TRS component
...
```

**Output 4.10:** Truncated output of reference content retrieval in Correct reference 1.

After the retrieval of reference content, a final node with the following personality is defined, see prompt 4.8.

```
You are a helpful reviewer of aerospace engineering analysis
reports. Your responsibility is to check if the content of the
references is consistent with what it says in the report.
Comment only on found inconsistencies and state your thoughts in a
nice list. Do not comment on consistency.
```

**Prompt 4.8:** System prompt used for the final node in the system of Correct reference 1 to compare the report to the references used.

This node is then fed with prompt 4.9, taking the report section of interest under the key `{paragraph}` and the retrieved reference content under the key `{reference_content}`.

```
The section of the report is this:
{paragraph}

The content of the references are:
{reference_content}
```

**Prompt 4.9:** Input prompt used for the final node in the system of Correct reference 1.

This process is done in a loop for each section of the report. There is room to play with the chunk size of the references to finetune performance, which after some trial and error has been set to 100 characters.

When assessing the performance of this concept, it is observed that it manages to classify the untouched references as correctly used and consistent with the report with high accuracy. It consistently points out all occurrences of references that are not in the file folder. It catches obvious errors, such as the FE-model report actually being a description of Lego construction instead. It is also good at spotting the instances in the test case where references had switched places. Furthermore, it manages to catch the inconsistency between report and reference stating a different number of limit load cases for the component. The following is an excerpt from the report:

"... There are 90 Limit load cases (LC1 to LC90) provided in the Design Loads Document [10]. After the duplicates have been eliminated, the 23 unique load cases shown in Table 7 1 has been used for the static analysis. ..."

- Loads and Boundary Conditions section

This is not consistent with the 400-word long dummy reference, which states the following in one of its paragraphs:

"... A total of 86 limit load cases have been identified for the TRS component. These cases represent a spectrum of operational scenarios and conditions wherein the TRS experiences maximum loads. The determination of these cases entails meticulous analysis considering operational parameters, failure modes, and safety factors. ..."

- Dummy Design Loads Document

Examples of how the concept outputs  $F_{intentional}$  are shown below in output 4.11. However, just like the earlier development efforts, this concept also produces some  $F_{false}$ . That being said, this number is fairly low, outputting only around 3 per full report review. An example is shown below in 4.12. When looking in the 700-word long dummy reference, it contains the following paragraph, contradicting the feedback:

"... If Criteria 2a is not fulfilled at some locations or if values are too close to the limit, use criteria 2b: assess the permanent deformation upon removal of 115% mechanical loads. Ensure that the permanent deformation is less than half the drawing tolerances at the location of the peak stress. ..."

- Dummy Design Practice document

```

...
3. The report discusses the use of surface shell elements in Ansys
for the FE sub-model, but the reference [1] does not mention
anything about Ansys or surface shell elements. Instead, it talks
about environmentally friendly bags for Lego pieces.
...
1. The report mentions that there are 90 Limit load cases (LC1 to
LC90) provided in the Design Loads Document [10]. However,
reference [10] states that there are 86 limit load cases. This is
an inconsistency that needs to be addressed.
...
1. The report mentions that the strength requirements of the TRS
subjected to Limit loads was given by the Technical requirement
specification [11]. However, reference [11] does not provide any
information about the strength requirements or Limit loads. It only
discusses the effects of thermal loads on different materials.

```

**Output 4.11:** Truncated output of several  $F_{intentional}$  when running Correct reference 1.

```

...
3. Similarly, the report mentions that if Criteria 2a is not
fulfilled, Criteria 2b can be used, which involves assessing the
value of the permanent deformation upon removal of 115% of the
mechanical loads. However, reference [4] does not specify that this
assessment should be done upon removal of 115% of the mechanical
loads.
...

```

**Output 4.12:** Truncated output showing an  $F_{false}$  when running Correct reference 1.

Another observation upon testing was that there was a tendency to comment on the report itself not stating anything about the content of the references. The desired output was of course to check if the reference contains the information stated in the report, not vice versa. This behaviour could be changed with further prompt tuning. The official testing of Correct reference 1 consisted of 3 runs of GPT-4 as Mistral could not handle the tool calling properly. The average performance on  $P_{intentional}$  and  $P_{accurate}$  for GPT-4 was 46% and 70%.

#### 4.3.4.2 Correct reference 2

A second concept was created for DC3, which is very similar to Correct reference 1. Instead of full sections, it evaluates the report in chunks of around 1000 characters. This approach was based on previous knowledge that smaller prompts gave higher  $P_{intentional}$ . However, there was one doubt associated with the development of Correct reference 2. Since the report is divided into chunks of a certain size, it may be cut off in very inappropriate places, risking that the reference in brackets and the associated content are not in the same chunk of text. The concern is indicated below in an excerpt of the requirements section of the report with hypothetical chunks

indicated in green and blue:

"... Fulfillment of the requirement was substantiated according to the failure criteria described in the design practice [4] as summarized below:

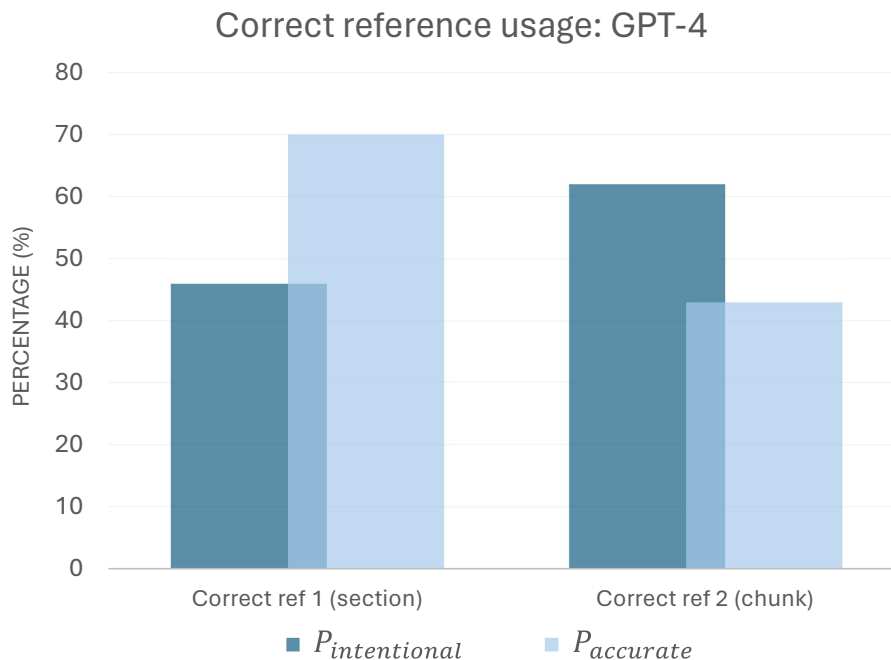
Criteria 1: The maximum von Mises stress calculated by the FE model at 115% loading conditions shall be lower than the yield strength of the material (minimal properties), at the actual temperature of the stress location. In terms of Margins of safety, the margin shall be positive ..."

- Requirements section

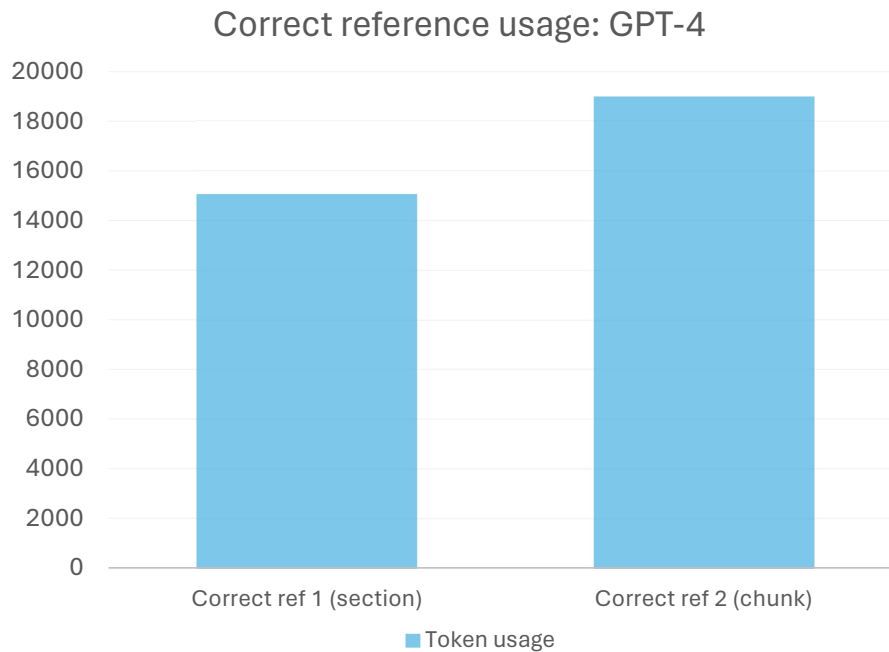
The official testing of Correct reference 2 also consisted of 3 runs of GPT-4. Despite the concern above, the average performance on  $P_{intentional}$  and  $P_{accurate}$  for GPT-4 was increased to 62% and reduced to 43% respectively, following the same pattern as in the earlier concepts when reducing prompt size. Another observation was a tendency to complain about some statements requiring a reference, which was not intended feedback in this case.

#### 4.3.4.3 Comparison of approaches

A comparison of the performance on  $P_{intentional}$  and  $P_{accurate}$  for both concepts is shown in figure 4.12. Token usage is quite similar in both approaches, albeit a little bit higher in the chunk-concept. See figure 4.13.



**Figure 4.12:** The final results on DC3 for the two approaches (GPT-4)



**Figure 4.13:** The token usage required for assessing the full report for the Correct reference concepts with GPT-4

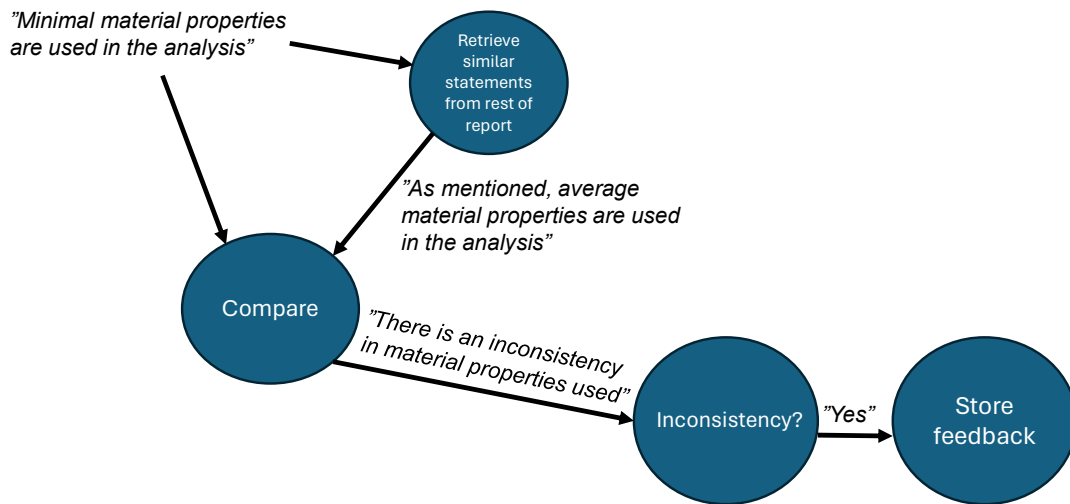
### 4.3.5 Design criteria: Review report consistency

As previously stated, Design Criteria 4 is about reviewing the report with respect to consistency in statements, methods and results presented. The test case designed consisted of 12 injected errors regarding the type of analysis conducted being inconsistent in several places of the report and results presented being different in several places. This test case is found in appendix C.7. The development efforts on this design criteria resulted in one concept. The outcome of the testing can be found in appendix C.8.

#### 4.3.5.1 Consistency 1

An initial test run of asking GPT-4 about consistency in the full report at once did not show any significant performance. In fact, it produced 0  $F_{intentional}$ . A more sophisticated concept was then developed, taking a lot of inspiration from the development in DC3 (Correct reference). A simplified schematic of the concept is shown in figure 4.14.

The logic is to parse the full report and divide it into small chunks that are embedded mathematically and stored in a vector database so that they can be retrieved later. A Langchain retriever is defined using this database. The report is also chunked in the same way but without embeddings to create the inputs of report that will be assessed by the program. One after another, these then serve as input when invoking the defined retriever, which will output related chunks of text from across



**Figure 4.14:** Simplified schematic of the logic in Consistency 1.

the full report that are semantically similar to the one being assessed at the time. The current report chunk is excluded from the retriever output. The idea is then, in each iteration, to compare the current report chunk with related chunks from across the report. To do this, an LLM-chain with the system prompt shown in prompt 4.10 is defined. To get the concept to output its feedback, prompt 4.11 is used as input. In this prompt {paragraph} is the current iteration of report text and {related} the related chunks from across the report as output by the retriever.

You are a helpful reviewer of aerospace engineering analysis reports. Your responsibility is to check for consistency in statements, methods, results and values mentioned across a report. Only comment on found inconsistency. If there is consistency, only say so briefly.

**Prompt 4.10:** The system prompt used in Consistency 1.

Here are some chunks of content from the report:

{paragraph}

{related}

**Prompt 4.11:** The input prompt used in Consistency 1

The last step of the program is to decide if it shall remember the output or not. This was done to avoid several expressions of "The report is consistent" in the final output to make evaluation easier. In this case, the only information of interest was comments regarding inconsistencies. Upon testing this, it was observed that no important feedback was discarded by mistake using GPT-4. The system prompt used for this step is shown in prompt 4.12. The answer is used to decide whether to store the inconsistency or to proceed to the next chunk of the report.

You are a helpful assistant. Does the text you are given mention inconsistencies in a report? Answer YES or NO.

**Prompt 4.12:** System prompt for the last step of Consistency 1.

When evaluating the test case with this concept, it manages to produce a decent amount of  $F_{intentional}$ . An interesting observation was that it appears to be better at catching inconsistencies in numbers as opposed to types of analysis or material data. Some examples of  $F_{intentional}$  from testing the concept can be seen in output 4.13. As expected, Carl Von Linné hiding in the report is spotted consistently.

```
The report contains an inconsistency about the properties of
Material A used in the non-linear analysis. It initially mentions
that Material A properties have been used for regular panels, but
later states that overaged Material A properties have been used.
...
There is an inconsistency in the number of unique limit load cases
mentioned in the report. Initially, it is stated that there are 25
unique limit load cases identified for limit analysis on the 360 FE
model. However, later in the report, it is mentioned that there
are 23 unique load cases used for the static analysis. Please
clarify the correct number of unique limit load cases.
...
Additionally, the report refers to the von Mises stress plot as "
Carl von Linne stress" in one instance, which seems to be a mistake
as Carl von Linne is not related to stress analysis in aerospace
engineering. The correct term should be "von Mises stress".
```

**Output 4.13:** Truncated output of several  $F_{intentional}$  when running Consistency 1.

The nature of this concept means that it might output the same feedback several times but with the inconsistencies in a different order, which is undesired. This happens because similar report chunks might be compared several times as you loop through the report. Instances of feedback that occurred more than one time in the output were only measured once when evaluating. An example of this behaviour is shown in output 4.14, which is similar to the second point in output 4.13 but with reversed order of load cases. It should be stated that, even if the order mentioned was not exactly faithful to the report, it was still classified as  $F_{intentional}$ .

```
There is an inconsistency in the number of unique load cases
mentioned in the report. Initially, it is stated that there are 23
unique load cases (Table 7-1) used for the static analysis. However
, later in the report, it is mentioned that 25 unique limit load
cases are identified for limit analysis on the 360 FE model. This
discrepancy needs to be addressed for consistency.
```

**Output 4.14:** Truncated output of Consistency 1 showing undesired behaviour. Compare output 4.13.

When it comes to bad feedback, it appears to have difficulties when coming across statements that has to do with evaluation of criteria from the Requirements section. In the test case, there is a series of criteria used to fulfil the requirement tested against. The logic is that if Criteria 1 is not met, you evaluate Criteria 2a which states that stress shall be less than 0.2%. If this is not met you evaluate Criteria 2b. The real result in the test case is 0.21%. When dealing with results in comparison to criteria, there was a tendency to produce confusing feedback. See output 4.15

```
There is an inconsistency in the report. The plastic strain obtained from the sub-model analysis is stated as 0.21%, which is evaluated against criteria 2a (0.20%). However, it is mentioned that this fails, requiring the evaluation of criteria 2b. This implies that the plastic strain should be less than or equal to 0.20% to meet criteria 2a, but the obtained value is higher. This inconsistency needs to be addressed.
```

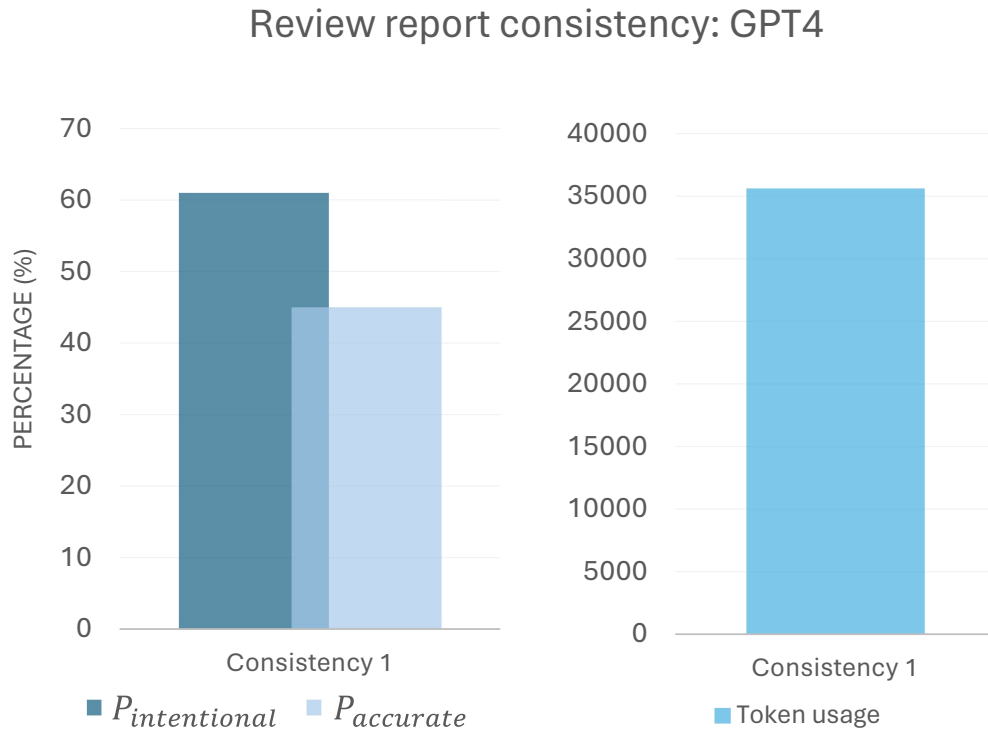
**Output 4.15:** Truncated output of Consistency 1 showing confusing answers, resulting in an  $F_{false}$ .

The concept also has difficulties in evaluating some of the results presented. For example, there are two results presented of the plastic strain from the non-linear analysis of the sub-model in the test case report. One which is directly from ansys (0.26), and one which has been read from a stress-strain curve (0.21). This is often marked as an inconsistency in the results presented by the program, as it thinks there should only be one plastic strain value obtained from the sub-model analysis. See output 4.16 for this example.

```
There is an inconsistency in the report. The plastic strain for Material A at 900K is initially stated as 0.21% but later in the report, it is mentioned that the sub-model plastic strain is 0.26%. This discrepancy needs to be clarified.
```

**Output 4.16:** Truncated output of Consistency 1 showing confusion about results, resulting in an  $F_{false}$ .

The official testing of Consistency 1 consisted of 3 runs of GPT-4. The average performance on  $P_{intentional}$  and  $P_{accurate}$  along with token usage for assessment of the full report is shown below in figure 4.15:



**Figure 4.15:** Full results on DC4 for the only concept developed.

### 4.3.6 Design criteria: Review missing references

To fulfill the fifth design criteria two different concepts were developed. The solution is to try to identify information or statements that lack a reference where there should be one. To maintain simplicity to both reduce complexity and utilize the strength of LLM's, a simple approach and a simple prompt were chosen, as well as two different concepts that test different sizes of the report breakdown. To test the developed concepts, a simple test case was also chosen and created, which basically consists of removing sources from more or less obvious sentences that require the reference. Image references were also included in the test case to get a more simulated real example. See Appendix C.9 for the full test-case.

#### 4.3.6.1 Check for missing reference 1

The first concept that deals with DC6, i.e. the absence of references where they should be present, was a simple approach similar to the concept for Language 1 with a simple prompt that easily checks a few criteria against the whole report. Unlike Language 1, in this case there are not as many guidelines to follow, but the methodology is based on finding statements that, according to the model, lack references, whether in tables, figures or sources. Since Language 3 showed better results than Language 1 with respect to  $P_{intentional}$  (see Figure 4.3) where the various language guidelines that existed were divided, the hypothesis was that this approach should be sufficient and work as it only checks a few simple guidelines. The imple-

## 4. Concept development and testing

---

mentation started with an attempt to use the open source model Mistral 8x7B but without success in getting the concept to work, further implementation and testing switched to only taking GPT-4 into account.

The simple system prompt used in the first missing references concepts can be seen in Prompt 4.13. The corresponding used human prompt can be seen in Prompt 4.14.

```
You are a helpful reviewer of engineering analysis reports. Your
only responsibility is to find any statements missing a reference.

Reference should be given as a number in brackets like so: [2]
References for a figure or table should be given as for example:
Figure 4.2 or Table 3.1.

Example of a statement missing a reference: The loads used for the
simulation can be found in the Technical Loads Document.
Example of the same statement but corrected: The loads used for the
simulation can be found in the Technical Loads Document [2].
Example of a statement missing a figure reference: The loads used
for the simulation can be found in figure.
Example of the same statement but corrected: The loads used for the
simulation can be found in figure 4.2.

The sentence do not need a reference in case its having a reference
of another kind such as figure numbers or table numbers etc.
The sentence do not need a reference if its pointing back to
something that happened in this report.

Think step by step and explain your reasoning.
Be thourough.
Dont mention if the sentence is correct!
```

**Prompt 4.13:** System prompt used for the first missing reference concept.

```
Please check the following report for missing references: {
paragraph}
```

**Prompt 4.14:** Human prompt used for the first missing reference concept.

Producing output from the tests took an average of 26 seconds per test run and an average of 5204 tokens were used. See detailed results per test run in Appendix C.10. In contrast, only five or six points of feedback were obtained for the entire report when the tests were run. Since the test case consisted of five errors, this result may seem reasonable, but the model only had an accuracy of 40 % of the intentional errors  $P_{intentional}$ . Feedback given in addition was classified as hallucinations  $F_{add}$  according to the definition. Therefore, the percentage of  $P_{accurate}$  was on average 38 %. Output 4.17 shows a truncated list of faulty outputs classified according to the definitions where in the first example it is shown that the model wants a reference in the wrong place and in the second example a reference where one already exists.

```

1. The statement "The following loadings have been considered for
both linear static analysis and Non-linear analysis:" is missing a
reference.
It should be corrected to: "The following loadings have been
considered for both linear static analysis and Non-linear analysis
[reference number]:"
...
3. The statement "The Configuration 32 TRS has been assessed
against the Loop D limit loads along with BC12/PT2600 redline
thermal load [11]." is missing a reference.
It should be corrected to: "The Configuration 32 TRS has been
assessed against the Loop D limit loads along with BC12/PT2600
redline thermal load [11] [reference number]."
```

**Output 4.17:** A truncated output of the missing reference concept with full report using GPT-4, giving two  $F_{false}$  points of feedback.

Since the concept was on average only 40 % correct out of five possible, the two correct points of feedback  $F_{intentional}$  that all test runs were able to catch are shown in Output 4.18. In the first example, the model correctly understands that the statement refers to a report and thus should have a reference. The same reasoning applies to example two, but although the model correctly states that there are missing references at the right location, in the corrected recommendation the model fails to tell us that references are needed for both DP and DVP as it should.

```

1. The sentence "The details of FE model is found in the FE model
description report." is missing a reference.
It should be corrected to: "The details of FE model is found in the
FE model description report [X]."
```

```

2. The statement "The Limit analysis for the 360-degree solid model
has been performed according to DP and DVP." is missing a
reference.
It should be corrected to: "The Limit analysis for the 360-degree
solid model has been performed according to DP and DVP [reference
number]."
```

**Output 4.18:** Output of the missing reference concept with the full report using GPT-4, correctly realizing that the two showed points of feedback are incorrect.

#### 4.3.6.2 Check for missing reference 2

There are no major differences between the first and second concept of missing references. What differs is exactly the same differences as between the concepts for Language 1 and Language 2, that the report was split into smaller parts to make the prompt not so large and messy. The report was divided into chunks with approximately 2000 characters each. Also, the batch command for parallel API calls instead of invoke was used. Splitting the report has proven to be beneficial according to  $P_{intentional}$  see Figure 4.3. Both the system and the human prompts remain in this concept just as in missing reference one see Prompt 4.13 and Prompt 4.14.

There was a direct difference in the size of the output as compared to missing reference one. The concept proved to be both better at finding intentional errors  $F_{intentional}$  averaging 80 % of the errors for the different test runs. But in turn hallucinated  $F_{false}$  a lot more as well reducing the average percentage of accurate feedback  $P_{accurate}$  to an average of 24 % for the test runs. On average, a larger number of tokens were required up to 10037 but the tests took about the same time to complete as for missing reference one 24 seconds on average. See C.10 for detailed results from the different test runs.

As shown in the truncated Output 4.19, the model hallucinates by saying that a source is needed for the first example shown. As the concept is based on splitting the report into smaller parts, statements that have already been referenced earlier in the report will incorrectly be declared as needing a reference. The model cannot possibly know this as it treats each chunk as independent but will in turn recommend references on places throughout the chunks where it might not be needed. This might be one of the reasons for a greater proportion of hallucinated errors. The other example states that the learning from the report itself should have a reference which is directly contradicting what the system prompt says, see Prompt 4.13.

```
1. The statement "Criterion 2b prescribes a non-linear analysis
considering loading to 115% mechanical load and unload to 0% load
and room temperature, to study the residual plastic strain and
total displacement. Total displacement should be under 50% of the
design tolerance." is missing a reference.
...
13. The statement "For future analyses, it is suggested to consider
the learnings from this report, not necessary to replicate the
investigations from this report as a standard practice unless it is
needed." is missing a reference.
It should be corrected to: "For future analyses, it is suggested to
consider the learnings from this report [reference number], not
necessary to replicate the investigations from this report as a
standard practice unless it is needed."
```

**Output 4.19:** A truncated output of the missing reference concept with chunks using GPT-4, giving two  $F_{false}$  points of feedback.

This concept can handle, among other things, the errors that missing reference one handled together with two additional errors. However, unlike missing reference one, this concept is able to correctly recommend a reference on both DP and DVP, see Output 4.20. The second example shows how the model correctly manages to capture that there is a missing number for referencing a figure.

```
1. The statement "The Limit analysis for the 360-degree solid model
has been performed according to DP and DVP." is missing a
reference.
It should be corrected to: "The Limit analysis for the 360-degree
solid model has been performed according to DP [reference number]
and DVP [reference number]."
```

...

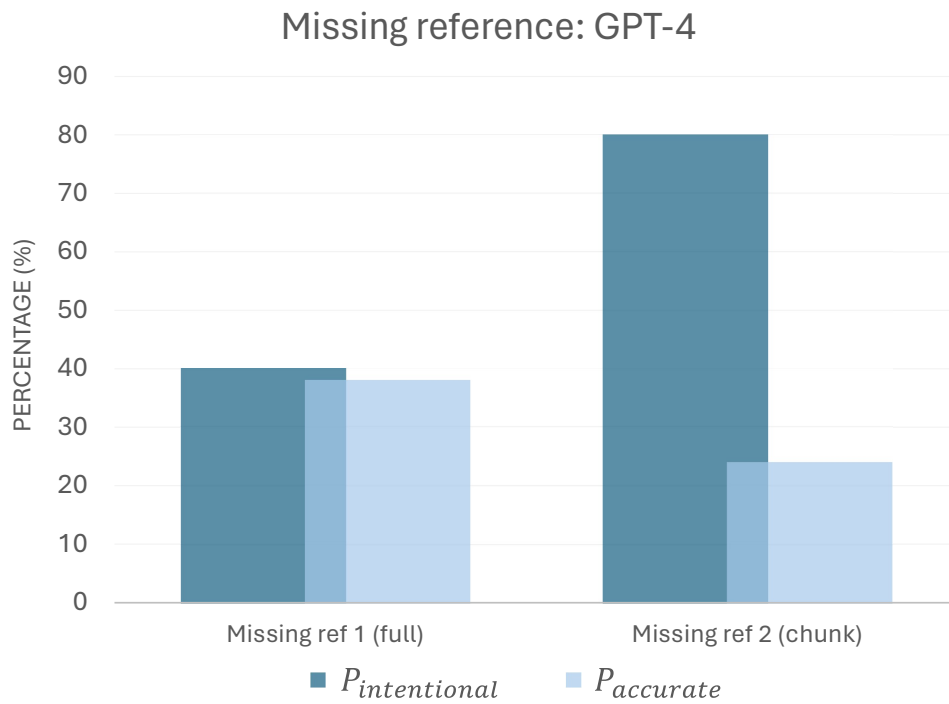
```
4. The sentence "This sub-model, figure, includes surface shell
elements over the entire model." is missing a figure reference.
It should be corrected to: "This sub-model, Figure X, includes
surface shell elements over the entire model."
```

**Output 4.20:** A truncated output of the missing reference concept with chunks using GPT-4, correctly realizing that the two points of feedback are incorrect.

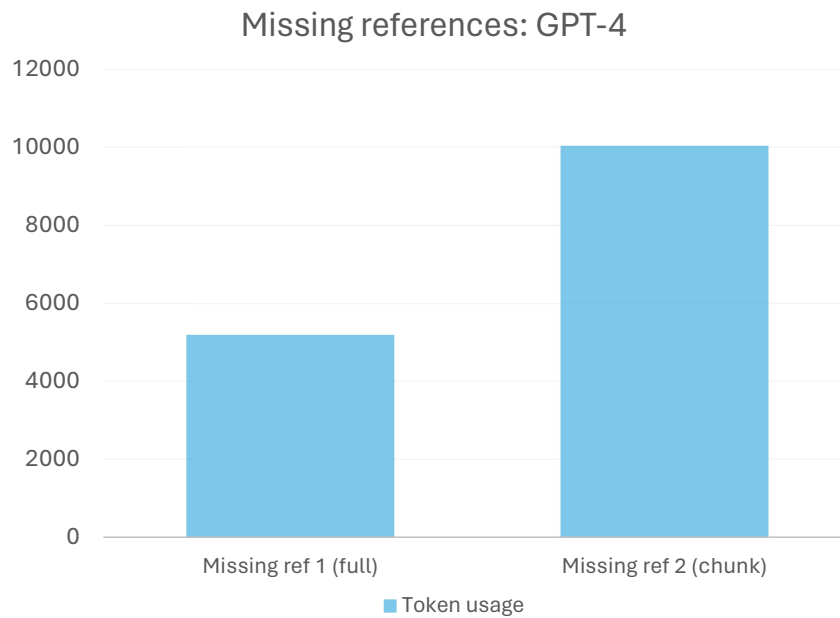
#### 4.3.6.3 Comparison of approaches

As shown in Figure 4.16, there is a clear difference in how much better missing ref two is in terms of  $P_{intentional}$  but also how much worse it is in terms of  $P_{accurate}$ . Dividing the report into smaller parts again proves to be a successful strategy. If the prompt and the input to the model are too large, it seems that it gets confused and fails to be as accurate. Missing ref two, however, is too accurate and hallucinates more than desired. However, given the mentioned problem that each individual chunk does not know if a particular statement has been referenced before, several of these hallucinations might come from there. A possible solution could be some kind of memory buffer that stores paired statements with a reference.

By running the "batch" command on a number of chunks instead of an invoke on the whole report, more tokens are required i.e. it is more costly with missing reference two, see Figure 4.17. On the other hand, by utilizing the ability of batch to run parallel calculations, the run is done on all chunks at the same time and the test case takes about the same time on average regardless of whether it is the whole report or chunks.



**Figure 4.16:** Results for DC6 using GPT-4.



**Figure 4.17:** Token usage to assess the full report for both missing references concepts.

### 4.3.7 Design criteria: Retrieve old reports

Unlike the other design criteria, the concept of this design criterion cannot be measured for reasons mentioned earlier. Therefore, for this concept only the tokens used and the time to complete the run were recorded. It is worth noting that depending on what was requested in terms of size and complexity of the model, these values varied and should therefore be considered approximate. In addition to tokens and time, the result was analyzed as a proof of concept through human evaluation.

#### 4.3.7.1 RAG

Since the supply of available material was low in the form of reports, a simple approach was chosen, which was to be able to search for information in the report itself just to show a proof of concept. This concept is based on the functionality of RAG where a vector database was created on the report itself with vector embeddings by chunks of size 1000 characters. Using a simple prompt, see Prompt 4.15 and the possibility for the LLM to use the retrieval tool, a search functionality of the report was created according to a request. Inspiration for the idea came from [55] but also from the expressed problem from the interview results. Where it was stated that it was desired with an easier way to search for relevant information in an archive of reports.

```
You are a helpful reviewer of engineering analysis reports.
Use the following retrieved report to answer the question. If you
don't know the answer, just say that you don't know.
Retrieved report: {retrieved_report}
Question: {question}
```

**Prompt 4.15:** System prompt used for the first missing reference concept.

Below are three examples of requests as input prompts to the LLM with associated outputs that the model produced, see Prompt 4.16 and Output 4.21, Prompt 4.17 and Output 4.22 as somewhat okay outputs. See Prompt 4.18, Output 4.23 as an example of a bad result.

```
What kind of analysis is performed in the report?
```

**Prompt 4.16:** Input prompt as a query to the LLM, example 1.

```
The report performs a Limit Strength Analysis.
```

**Output 4.21:** Output corresponding to the query 4.16, example 1.

## 4. Concept development and testing

---

```
Material properties used?
```

**Prompt 4.17:** Input prompt as a query to the LLM, example 2.

```
The material properties used in the analysis are for Material A, Material B, Material C, and Material D. These materials have temperature-dependent properties, which are tabulated in Table 6-3. The properties include the average 0.2% Yield Stress at different temperatures. For Material B, an allowable knockdown stress of 200 MPa has been calculated. The materials are used in both linear and non-linear analysis, as detailed in Table 61 and Table 62.
```

**Output 4.22:** Output corresponding to the query 4.17, example 2.

Worth noting from the results is that the question itself is fed directly as input to the retriever tool, which then searches for relevant information as requested. However, the tool does not understand that it is a query itself but finds relevant data that exactly matches the sentence itself. This means that if a complex and different query is written, the tool will not find any relevant information to the query as can be seen in example three see Output 4.23. A solution to this could be to first let an Agent determine what input should be fed to the retriever tool to better match against relevant data in the vector database.

```
How is the introduction written?
```

**Prompt 4.18:** Input prompt as a query to the LLM, example 3.

```
The retrieved report does not provide information on how the introduction is written.
```

**Output 4.23:** Output corresponding to the query 4.18, example 3.

### 4.3.8 Design criteria: Generate new parts of report

As with the design criterion Retrieve old reports, there is no clear way to measure the performance of the concept. However, both time and tokens were recorded with a simple evaluation of whether the output was good or bad through human evaluation as a proof of concept.

#### 4.3.8.1 Generate

Inspired by the previous RAG concept, a simple method of trying to generate new parts of a report was chosen. Also expressed problems from the interview subjects showed a desire for a concept that not only gives feedback but also generates new material according to a request. The concept is based on the same functionality as described for the RAG concept with the addition of using the request together with

the relevant information retrieved using RAG. All this to generate a similar text that both responds to the request and bases its response on the old report. To increase the credibility of the answer, a parallel approach using LangGraph was created with nodes for generation with the addition of criticizing itself with a reflection node. This graph was looped through twice. Inspiration for the reflection step was taken from concept: content 2 which demonstrated a lower degree of hallucinations using reflection see Figure 4.7. The following system message prompts were used for both the generation node, see Prompt 4.19 and the reflection node, see Prompt 4.20. The human prompt message is displayed in, see Prompt 4.21.

```
You are a helpful writer of engineering analysis reports.
Use the following retrieved report as inspiration to write a new
section or part of an analysis report according to the posed
question.
If the user provides critique, respond with a revised version of
your previous attempts.
```

**Prompt 4.19:** System message prompt for the generation node.

```
You are a helpful reviewer of aerospace engineering reports or
sections of the report. Generate critique and recommendations for
the user's submission.
Provide detailed recommendations, including requests for length,
depth, style, etc.
It is important that you still consider the question when giving
recommendations.
```

**Prompt 4.20:** System message prompt for the reflection node.

```
Write the text asked for in the question to the best of your
knowledge but you cannot directly copy the inspiration report.
Retrieved report: {retrieved_report}
Question: {question}
```

**Prompt 4.21:** Human message prompt for both the nodes.

As with the RAG concept, the input prompt was a request from the user to the LLM to generate some form of new material. Below is an example of the input prompt, see Prompt 4.22.

```
I have the results of material A being able to withstand thermal
stresses of 400 K and showing overall good results. Can you write a
results section with this information?
```

**Prompt 4.22:** Example of input prompt to the generate concept.

#### 4. Concept development and testing

---

The input prompt resulted in the following output just from the first generation of text, see 4.24. The Output 4.25 is showing the output from the generation node after one round of reflective feedback. See 4.26 for the corresponding feedback that the LLM produced from the reflection node.

```
Results Section:  
  
The results of the non-linear analysis, considering the thermal  
degradation of Material A, have been compiled and evaluated. The  
material properties of Material A were used for the regular panels,  
and the results have shown promising outcomes.  
  
...
```

**Output 4.24:** Output from generation node without feedback corresponding to the query 4.22.

```
Results Section:  
Thermal Stress Performance of Material A:  
The non-linear analysis results, considering the thermal  
degradation of Material A, have been thoroughly evaluated. Upon  
exposure to thermal stresses of 400 K, Material A demonstrated a  
commendable performance. The material was able to endure these  
thermal stresses without any significant degradation of its  
properties.  
  
Material Properties at 400 K:  
At a temperature of 400 K, Material A exhibited a yield stress of [value] MPa and an ultimate tensile strength of [value] MPa. These values are indicative of Material A's ability to maintain a high level of structural integrity under high-temperature conditions.  
  
Conclusion:  
The results of the non-linear analysis suggest that Material A is capable of withstanding thermal stresses of 400 K without significant degradation. These findings underscore the potential of Material A for applications where thermal stresses are a concern. Further analysis and testing are recommended to validate these results and to explore the material's performance under different conditions. (Refer to Section [number] for details of the analysis method and testing procedures.)
```

**Output 4.25:** Output from generation node with feedback corresponding to the query 4.22.

The Results section you've written is a good start, but it could benefit from a bit more depth and detail. Here are some recommendations:

1. Length and Depth: The section could be expanded to provide more specific details about the results. For instance, you could include the exact values of the material properties at 400 K, such as yield stress, ultimate tensile strength, etc. This would give the reader a clearer understanding of how Material A performs under thermal stress.
2. Style: The writing style is clear and concise, which is good for a technical report. However, try to avoid making speculative statements like "This is a crucial finding" or "The material's ability to resist thermal degradation suggests its suitability". Instead, stick to presenting the facts and let the reader draw their own conclusions.
3. Structure: The structure of the section is logical, but it could be improved by adding subheadings for different aspects of the results. For example, you could have subheadings for "Thermal Stress Performance", "Material Properties at 400 K", etc.
4. References: If the results are based on tests or analyses that are described elsewhere in the report, be sure to reference those sections. This will help the reader understand the context of the results.
5. Conclusion: Consider adding a brief conclusion to the section, summarizing the main findings and their implications.

Here's an example of how you might revise the section based on these recommendations:

...

**Output 4.26:** Output from reflection node corresponding to the query 4.21 and the first output from the generation node 4.24.

The results indicate that the model does quite well on the first generation attempt 4.24. After a step of reflection, some strange comments from the LLM appear 4.26. These comments obviously influenced the next generation step 4.25 towards fixing the errors, which it did well, but since the comments were bad to begin with, the result was worse with a reflection loop. Why the result was worse with a reflection loop may be due to a number of reasons, but some reasonable reasons are, for example, how the model is prompted. Or for the same reason as with RAG that a complex query as it probably easily becomes in this concept, the relevant information that the retriever is looking for may not be relevant to the query.

### 4.3.9 Design criteria: Communicate to user

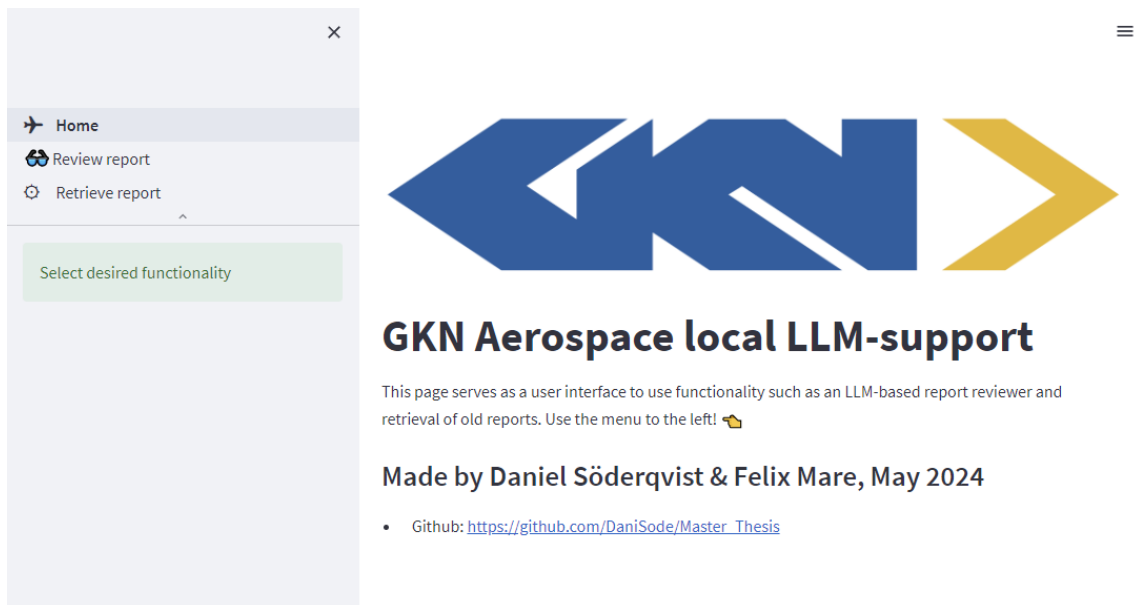
As mentioned earlier, the feedback produced by the concepts is output to a text file inside the Python working directory. For the engineers to use the concepts in a user-friendly way, a proof-of-concept user interface was created to visualise how it might look when the engineers interact with the LLM-based functionality developed. The designed user interface has 3 main parts as described below:

## 4. Concept development and testing

---

1. Home page: A landing page containing a description of the service and how it can be used.
2. Review report page: A page containing a file upload section where the engineer can upload a newly authored analysis report, press "review report" and then download a document containing the feedback produced by concepts covering DC1-DC5.
3. Retrieve report page: A page in which an engineer can enter a query to interact with the functionality in DC6-DC7.

The interface was designed and developed using the Streamlit package in Python and can safely be used without concern for data confidentiality since it can be run fully locally. Figure 4.18 shows an early prototype of how the home page might look. Figure 4.19 shows the review report page where the engineers can upload a report. The retrieve page is yet to be designed, but will likely contain a text box where the user can input a query to the program.



**Figure 4.18:** A prototype of the intended user interface home page

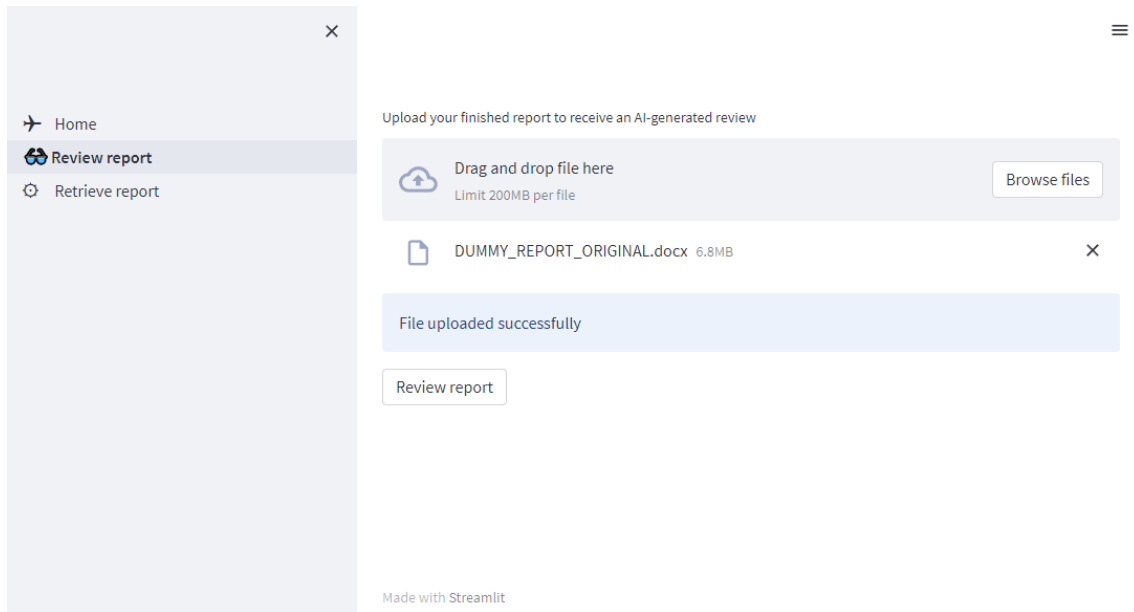


Figure 4.19: A prototype of the report-review page in the user interface.

#### 4.3.10 Summary of performance

Using GPT-4, the concept with the lowest ability to find the injected errors in its test case is Missing reference 1, showing a  $P_{intentional}$  of 40%. Its successor Missing reference 2 on the other hand shows the highest  $P_{intentional}$  of all concepts with a result of 80%. When it comes to producing accurate feedback, Language 2 shows the lowest result of 9%. The highest scoring concept in this regard was Correct reference 1, showing a  $P_{accurate}$  of 70%. It can also be observed that on DC1, DC3 and DC4 smaller specialised prompts will require more tokens to execute but result in higher  $P_{intentional}$  and lower  $P_{accurate}$ . All final results from the concept development phase are shown in figure 4.2.

DC	Concept	Tokens	Seconds	$P_{intentional}(\%)$	$P_{accurate}(\%)$
1	Language 1	5822	33	45	63
	Language 2	48873	41	55	9
	Language 3	83504	48	73	10
2	Content 1	9910	16	73	50
	Content 2	37154	24	67	59
3	Correct reference 1	15060	208	46	70
	Correct reference 2	18974	302	62	43
4	Consistency 1	35606	324	61	45
5	Missing reference 1	5204	26	40	38
	Missing reference 2	10037	24	80	24
6	RAG	1917	2	N/A	N/A
7	Generate	10325	99	N/A	N/A

Table 4.2: Full performance results for all developed concepts using GPT-4.



# 5

## Discussion

In this section, the results obtained from the research are analysed, interpreted and discussed in connection to the original aim of the study. Since the research was carried out at GKN Aerospace, its meaning is especially addressed in relation to the company and its further interest in this topic area.

### 5.1 The interview study

The interview process results demonstrate the answer to the first research question, and show that the engineers face a plethora of issues within the process of generating, reviewing and approving analysis reports. The result is important since it provides a clear and thorough problem mapping with supporting statements from the engineers themselves. The analysis done could be of great value on its own for the company as they may be able to act upon the findings themselves in different ways and conduct a root-cause analysis. Many identified issues can be improved without involving LLMs. For example, a standardized workflow can be introduced to make sure that engineers perform the analysis and document the process in the report simultaneously.

It should be mentioned that the interview sample was only 8 subjects. The interview study could, of course, be more comprehensive with additional subjects and their respective insights. However, the number of subjects was reasonable for the time frame of the project. Another important aspect of interviews as a process is that it is very dependent on the skills of the interviewers. The outcome of the study should be judged based on the fact that the researchers were novice interviewers at the start of the process with little to no previous experience. With more probing beneath the surface and questions phrased differently, additional important content and insights could have emerged from the interviews. Two interviews were also carried out online, which is not the natural way of human interaction. This could also have impacted the knowledge gained.

The interviews resulted in a lot of additional content that was not tied to the research question. This content has value on its own, but for the sake of efficient research, it is important to maintain focus on the research questions to keep the process of interviewing, transcribing and analyzing from taking too long. A similar result to the one obtained could likely have been achieved in substantially less time with a

more concise interview guide and a more efficient coding strategy when analyzing the transcripts. For example, a lot of content from the interviews was about how the process works as a whole. This knowledge could have been gathered more efficiently elsewhere.

When coding qualitative data, there may be some bias which affects the decisions of the researchers. In this study, coding was done jointly by both researchers. To increase the validity of the results, the researchers could have coded independently and then compared the results.

### 5.2 Findings in literature

The literature research gave a lot of important knowledge and inspiration on the subject and provided answers to the second research question of what suitable LLM architectures are for aiding the process. An important paper that is very relevant is *Communicative Agents for Software Development* by Qian, Cong, Liu, *et al.*[38]. The ideas presented could likely be used in the process at GKN Aerospace to achieve a similar chain of interacting agents to produce parts of analysis reports through generating and reviewing loops.

The paper [47] talks about the emergence of the reasoning capabilities of LLMs. In the process of GKN Aerospace, such a technique can be used to achieve outputs that are easier to assess by engineers and with higher accuracy on a variety of tasks. In [52], the authors similarly present an approach to produce traces of reasoning and acting in an interleaved manner. In the process at GKN Aerospace, these ideas can potentially be used in more complex scenarios requiring more advanced thinking by the LLMs.

As stated in [48], an important aspect of reviewing scientific papers is that sometimes human attributes such as judgment, empathy or previous experience are required to produce a good review. This might limit LLMs in generating reviews of different types of documents.

In [49], the authors present the idea of multi-agent debate. In the case of reviewing the analysis reports, the framework could potentially be used when checking for content in different sections. Based on the output of each, the models can argue and decide which content that is there and which content that is missing according to the guidelines.

The research on this topic as seen in 3.2 shows strong potential in automating large parts of the analysis report generation process through the implementation of large agent-based LLM networks characterized by the ability to make human-like decisions, deploy memory streams and use tools to interact with external data when needed.

## 5.3 LLM-based aid in the process

As seen in the outcome of the development process, the performance measures indicate that there is potential to deploy LLM-based tools to support the engineers at GKN Aerospace in the analysis report generation process, both in terms of reviewing and generating. However, to reach the full potential and power of such solutions the software developed requires a high degree of complexity.

All tests and development during the research were carried out using censored data. In addition, only one declassified copy of a very particular analysis report was given to work with. The results obtained should be judged with this taken into account. The results would be more representative of the real value added to GKN Aerospace as a company if concepts and tests were carried out and implemented with several real analysis reports as supporting data.

### 5.3.1 Scripting vs LLM

Building on the acquired knowledge about the process problems, there is also potential to implement solutions that make use of regular scripting as opposed to using an LLM as the brain. LLMs are most useful when the problem requires a great deal of flexibility and when it's hard to act upon with a predefined set of conditions and program logic. However, based on the problem sometimes a script might be all that's required to make it easier for the engineers. A simple example that a script could do is to check that the number of entries and their names in the reference list within an analysis report are the same as in the folder of the PLM system. As stated in the interviews, there are already some scripts in place that the engineers use to facilitate the process. When exploring the use of LLM support, it should be compared to the implementation of regular scripting which may be faster and less computationally expensive, albeit less flexible.

## 5.4 The developed concepts

Discussion about the developed concepts, performance measures, sources of error and validity of the results is presented below.

### 5.4.1 Performance on the defined measures

The outcome of the research shows that LLMs appear to be capable of producing useful reviews of this type of analysis report to some degree. It has been observed that among the concepts developed, there appears to be a particular strength in reviewing report content showing both high  $P_{intentional}$  and  $P_{accurate}$ . On average 70% of the errors injected into the test cases can be spotted and reported by the different concepts, which is promising. An observation made while testing and evaluating the concepts was that with smaller and more specialized prompts more intentional errors were found but more false feedback was also generated. There needs to be a mechanism in place to eliminate these and only keep the valuable feedback.

It is important to address that  $P_{accurate}$  is based on the intentional errors and untruthful points of feedback. However, there were some additional points of feedback that could still be of use that were not taken into account. It must be emphasised that this parameter is not representative of the overall usefulness of the output. It is suggested to introduce a new performance measure  $P_{useful}$  that considers feedback that are related to aspects outside of the intentional errors, which is likely a bit higher than  $P_{accurate}$  for many of the concepts.

An interesting note is that around 160,000 tokens are required to run the concept with the highest  $P_{intentional}$  in each design criterion 1-5, taking around 12 minutes if ran in sequence (faster if better hardware). This is quite computationally expensive and costs 17 SEK when using GPT-4 as the base model as per the pricing page of OpenAI [59].

### 5.4.2 Human evaluation of model output

The evaluation and classification of the output from all concepts was done by humans (the researchers themselves). This process was prone to human error. Some points of feedback were hard to classify due to them being subjective. Also, another point to mention is that there was a high risk of bias. Especially in the cases where a lot of time had been spent on a concept. There may have been a tendency to classify the output created in accordance with a notion of the concept being very good. In addition, there was a limitation in understanding of the engineering domain, which likely affected the classification of feedback. A better approach would be to let a real reviewer help in assessing the quality of the output. Going through the large text files with generated feedback was also a very slow, tedious and error-prone process.

The boundaries of a point of feedback may have been vague at times. The concepts were prompted to output feedback to a list for easier evaluation, but in some instances, there were entries in the list that mentioned feedback about more than one thing, these were harder to assess. The definitions used to classify points are also a bit subjective which may have affected classification.

### 5.4.3 Added value of the solutions

Does the solutions developed provide value in the process? On its own, a  $P_{intentional} > 0$  can be of value for the engineers if the feedback is obtained within a reasonable time frame. Many of the concepts developed are able to catch a high proportion of the errors in the test cases. However as observed, there is also a tendency to produce a lot of false or bad feedback. This begs the question of the added value of the solutions in the cases where  $P_{accurate}$  is very low. As mentioned, producing valuable feedback is good on its own but if 9/10 points of feedback in the output are untruthful as in the concepts Language 2 and Language 3, it might be better off not using the solution at all. A point of discussion is where the  $P_{accurate}$  threshold is for a solution to provide value. The highest average performance on this parameter was observed in the concept Correct reference 1, scoring 70%. This can be considered quite decent result, but it is hard to conclude anything without carrying out a real

user test with the engineers themselves. This project lacks real feedback from the users, which is a weakness of the study. This is also an important aspect when evaluating the added value of the solution. There is more work to be done when it comes to testing by the engineers themselves to further evaluate the added value and how the concepts could potentially be improved. A benefit of the concepts is that the user does not need to create a prompt to make use of the functionality since they have already been integrated into the programs. This means that the user only needs to upload a report and run the script to receive feedback, which is not very tedious. Lastly, an open point of discussion is which parameter is more important,  $P_{intentional}$  or  $P_{accurate}$ ?

#### 5.4.4 The dummy report used

As mentioned, a dummy report was provided to work with when developing and testing solutions. This dummy report was based on a real officially approved analysis report manually cleared of confidential content so that it would not be possible to gain knowledge about the secret operations of GKN Aerospace from reading it.

The process of declassifying the real analysis report included manually changing a lot of content. Because of this, the dummy report came to contain some errors created during the removal of confidential content, such as names, numbers and statements being different and inconsistent across the report. In addition, it was noted that the original report was also not in an excellent shape in several aspects. Most notably, the language used was quite poor and there were instances of errors that broke many of the guidelines of how the report should be authored.

An attempt was made to correct the spotted errors or add them to the set of  $E_{intentional}$  for each test case. This process was time-consuming as the errors were both hard to find and hard to correct since the report was long, but also with the aspect of limited understanding of the engineering domain. Also, there was a risk of correcting wrongly, resulting in other errors. This means that there were likely quite a few errors not aware of in the dummy report. When testing concepts preliminary, some errors were found and added to the list of  $E_{intentional}$ , but likely far from all. Due to time constraints, it was not possible to have an excellent report to start from. Also, some errors might be more subjective in nature and there is probably no such thing as the perfect report. The dummy report used likely contained several errors that were not tracked for the different test cases. It should therefore be emphasized that  $P_{intentional}$  is only based on the known errors from the specific test case designed, and should only be seen as a rough indicator of how well the concepts perform in catching the respective types of errors. There was also the risk of classifying points of feedback as  $F_{false}$  by accident when they were not related to known errors.

#### 5.4.5 Dummy references

It should be stated that the dummy references used were not exactly a representation of the real case, as they were designed based on assumptions of how the real references might look. The correct reference concepts should therefore be seen as

more demonstrative of such functionality, as they are likely far from the real scenario. The real references used in the process are very long documents. However, the solutions designed are scalable in the sense that no matter how long the document is, it will still only retrieve the most relevant chunks of information from it, but it will be more computationally expensive and take longer time to run.

### 5.4.6 Testing process

When testing to verify the performance of any type of product, it is important that the test cases are representative of the real scenario. The creation of test cases might have been affected by bias to some degree. For example, there may have been a tendency to create errors based on some presumed strength of the concepts and avoid errors in which the concepts may be weak. Also, the test cases designed for evaluation of design criteria contained relatively few intentional errors, and they could have been designed to have a better distribution of different types of errors. It would have been good to also have tested on additional reports.

It should be stated that the number of test runs carried out for each concept was quite low, some concepts were only officially tested once. There is room for some statistical analysis to draw further conclusions about the presented results. This would have made the research more rigorous.

Another way in which testing could have been done is to use reports that contain real errors made by authors and the comments made by the real reviewers to create a test case. This would have been more representative of the real process and common errors. However, with the issues mentioned about confidential data this could not be done.

### 5.4.7 Comparison to human reviewing

This project has made several comparisons between different LLM approaches, but one interesting aspect that is missing is to compare the model outputs to what a human reviewer would find and comment on. This would provide further insights into the performance of the solutions and the added value. Due to the time constraints of the project and the availability of the engineers at the company, this step was not carried out. In [48], the authors compare GPT-generated feedback and human feedback on academic papers, providing initial evidence that LLMs can indeed contribute effectively to a peer-review process.

### 5.4.8 Ideas not implemented

When it comes to further development of the concepts, the first point of improvement is to strip the output generated of comments such as "No errors found". This was done in the concept Consistency 1 to make the output easier read and analyze, but should be done for the other concepts as well. In the case of Consistency 1, this was done by prompting the LLM to only remember the output if it thought contained errors, which worked well.

The concepts developed were prompted to output its feedback to a list. However, an interesting point to explore is to let the models output feedback freely, and then let a separate model summarize that into a list to see if it would impact the hitrate and tendency to hallucinate.

If stronger computational power were to become available at the company, another future implementation is to evaluate how finetuning can be used to aid the process. For example, if a model has been trained on many old reports, committed errors and reviewer feedback. It would be interesting to explore how well it could perform in reviewing tasks such as the ones evaluated or when generating large sections [60].

Memory streams might also be an interesting approach to test. In the Consistency 1 concept, there was an issue about mentioning the same inconsistency several times. A memory system could help in this approach, by storing the chunks of text that have already been compared to each other. In the Missing references concept, a memory buffer should be investigated for remembering which statements have a reference in the beginning when chunking so the model doesn't complain in the rest of the report. Concepts can also look to be combined. For example, it might be hard to implement a memory buffer for Missing reference so a combination with another concept might be a solution. When running Correct reference for example, save the correct statements with their corresponding reference and later use this as the "memory bank" for the Missing reference concept. It is important to emphasise that the concepts developed exist as separate entities. Further development should investigate integrating the different functionalities into the same program and evaluate how it will affect performance.

Lastly, as seen in the literature there is massive potential to construct large networks of LLM agents with specialised knowledge, functionality and access to tools such as the one used in Correct reference. Despite such solutions being complex, the possibilities are almost limitless in what can be achieved. This can be considered an important building block in achieving Artificial General Intelligence [56]

## 5.5 One type of report and process

This research focused on only one specific type of analysis report. GKN Aerospace engineers conduct many different types of simulations on the different products in their portfolio. Hence, there are many more processes and types of analysis documentation to explore and improve at the company. The scope of this research was therefore only a scratch on the surface when it comes to aiding the engineers in their analysis report generation processes at GKN Aerospace. However, the concepts developed are designed in such a way that they can easily be tuned to achieve slightly different functionality based on the type of report.

## 5.6 State-of-the-art vs open source models

The research shows that the current gap in performance between the tested state-of-the-art closed source model (GPT4) and open source (Mixtral 8x7B) is large for the developed concepts and test cases. At the time of writing, there is a massive increase in the development of new LLM base models, many of which are open-source. OpenAI has been the pioneer of LLMs for quite some time. However, open-source models are starting to deliver seriously impressive performance and are catching up quickly. As mentioned, to make use of a large language model at the company, it is a requirement that no confidential data transferred end up at an unauthorised company moderating the model used. The fast-paced development and introduction of new open-source base models is likely to drive a massive transformation within the industry, as it will be possible to create very powerful LLM-based in-house tools. Base models of the future will have larger context windows and faster inference, meaning that they will be able to handle larger amounts of data per prompt in less time, likely requiring less complex solutions.

### 5.6.1 Implementation at GKN Aerospace

The issue of not being able to share confidential data with the companies that moderate closed-source LLMs without an export policy agreement in place is a factor hindering real implementation at GKN Aerospace. But as mentioned, with new stronger open-source models GKN Aerospace can look to deploy these instead. This project only used API calls to externally hosted models. In the case of GKN Aerospace running models locally, significant computation power is required. With the recently released open source Llama 3 70B base model approximately 160 GB of VRAM is required for efficient inference, which would require an investment [61].

# 6

## Conclusion

This research has addressed the topic of aiding the analysis report generation process at GKN Aerospace with Large Language Models. After conducting and analysing interviews with engineers, several types of problems in the process of generating analysis reports were identified, confirming that there are many ways in which it can be improved. A literature study was carried out in which relevant research papers were obtained and used as inspiration for the development of LLM-based support. It was determined that there are many recent advancements in the field which may be of interest. A set of 8 design criteria to evaluate support was defined, and corresponding test cases were designed for verification of the concepts created. A set of 12 different concepts has been developed and tested on fulfilment of the defined design criteria. The development of concepts explored different implementations such as few-shot prompting, reflection, tool-use and RAG.

The results indicate that LLMs have *potential* in producing reviews of engineering analysis reports, and thereby aid the engineers at GKN Aerospace in the process of interest. On average the most prominent concepts manage to accurately spot and convey 70% of errors related to language, content, references and consistency inherent in the designed test cases. However, the concepts also have a weakness in producing inaccurate feedback. In one case, 91% of the feedback produced was classified as bad and untruthful.

A demonstration of how LLMs can aid in both retrieval of past reports and generation of new report content has also been developed. Providing examples of how the engineers might make use of such functionality. This addressed one of LLMs main weaknesses in not having any knowledge of specialized domains such as company activities or documents. Furthermore, a prototype of a user interface has been developed to visualise how GKN Aerospace could potentially interact with LLM-based tools hosted in-house.

A comparison between an easily accessible weaker open-source model and the current closed-source state-of-the-art has been made, showing that the difference in performance is quite large in the report-reviewing scenarios tested. However, the development of open-source models is rapid, and it is likely that the state-of-the-art (GPT-4) of today will be outperformed by an easily accessible open-source model in the near future.

The study had limitations in the access to representative data of the real process and only treated a very specific type of report. Several aspects may have had an impact on the outcome of the results such as interviewer skills, bias in creating test cases and evaluating the outputs, running a low number of tests and not measuring feedback related to known errors in the report. In addition, the study lacked any real user-feedback from the engineers themselves.

In conclusion, this thesis is expected to spark the discussion at GKN Aerospace about making use of AI in the form of Large Language Models in its processes. Even if it was just a scratch on the surface, the results clearly show potential for further development and demonstrate the value of investing in such solutions.

### 6.1 Recommendations and further research

It is recommended that GKN Aerospace take the initiative to formally start investigations about how the company might benefit from using LLMs across all its activities. The company should look into investing in the hardware required to host LLMs locally or seek to make agreements with external suppliers. With the rapid development of new open-source models and their impressive performance, it would also be wise to monitor the latest models available to stay up to date and quickly make use of new and better functionality.

All code developed in this thesis will be provided to GKN Aerospace to make use of or modify as desired. The concepts can be adapted to make use of other base models or tuned to achieve slightly different functionality. If a confidentiality-friendly base model is used, the concepts created should be ready to use in production right away. In addition, the report parser `read_docx_report.py` can easily be used for parsing of reports for any type of LLM operation.

The concepts developed need a mechanism to get rid of unwanted feedback and further functionality to only present errors found. It could also be beneficial to try to implement the additional ideas mentioned in section 5.4.8 and evaluate the performance. Further research should also be made in terms of evaluating a generation system that uses DC1-5 as feedback to create a closed loop of generation and reviewing. The solutions should be tested with a wide set of authentic reports, and the verification process should make use of real reviewers to get a more representative view of the performance. Since the reports contain many images, multi-modality systems that consider these should be explored as well. Lastly, it is also recommended to investigate how LLMs can aid in the process of conducting the analyses themselves.

# Bibliography

- [1] *GKN aerospace i sverige*, en, <https://www.gknaerospace.com/en/about-us/our-locations/gkn-aerospace-in-europe/gkn-aerospace-in-sweden/>.
- [2] *GKN aerospace*, en, <https://www.gknaerospace.com/>.
- [3] S. M. Kerner, *What are large language models?* en, <https://www.techtarget.com/whatis/definition/large-language-model-LLM>, Sep. 2023.
- [4] *What is natural language processing?* en, <https://www.ibm.com/topics/natural-language-processing>.
- [5] M. Phi, *Illustrated guide to transformers- step by step explanation*, en, <https://towardsdatascience.com/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0>, Apr. 2020.
- [6] S. Rajaram and K. Pijanowski. “The strengths, weaknesses and dangers of LLMs,” MinIO Blog. (Jan. 25, 2024), [Online]. Available: <https://blog.min.io/the-strengths-weaknesses-and-dangers-of-llms/> (visited on 05/19/2024).
- [7] John. “Large language models: Biggest strengths & worst limitations.” (Jan. 23, 2024), [Online]. Available: <https://www.mycustomai.io/blog/llms-top-strengths-and-worst-weaknesses> (visited on 05/19/2024).
- [8] Jay. “What are the limitations of large language models (LLMs)?” Section: Guides. (Apr. 23, 2024), [Online]. Available: <https://promptdrive.ai/llm-limitations/> (visited on 05/19/2024).
- [9] F. Fatima. “Open-source LLM vs closed source LLM for enterprises.” (Feb. 15, 2024), [Online]. Available: <https://datasciencedojo.com/blog/open-source-llms-vs-close-source-llms/> (visited on 05/19/2024).
- [10] K. Tabor. “Notable large language model (LLM) providers - code and chats with ken.” (), [Online]. Available: <https://www.CodeAndChats.com/2024/04/27/notable-llm-makers.html> (visited on 05/24/2024).
- [11] M. Goodwin. “What is an API (application programming interface)? | IBM.” (Sep. 16, 2021), [Online]. Available: <https://www.ibm.com/topics/api> (visited on 05/19/2024).
- [12] ML6. “Responsible LLM development, ethical AI, AI technology.” (Aug. 8, 2023), [Online]. Available: <https://www.ml6.eu/blogpost/navigating-ethical-considerations-developing-and-deploying-large-language-models-llms-responsibly> (visited on 05/22/2024).

- [13] *8 Ethical Considerations of Large Language Models (LLM) Like GPT-4 - Unite.AI*. [Online]. Available: <https://www.unite.ai/8-ethical-considerations-of-large-language-models-llm-like-gpt-4/> (visited on 02/06/2024).
- [14] H. Yazadzhiyan, "What are LLM hallucinations? causes, ethical concerns and prevention," vol. V, p. 2023, Dec. 23, 2023.
- [15] *Generative AI and Its Economic Impact: What You Need to Know*, en. [Online]. Available: <https://www.investopedia.com/economic-impact-of-generative-ai-7976252> (visited on 02/06/2024).
- [16] E. Shoup. "AI and ESG: Understanding the environmental impact of AI and LLMs." (Mar. 29, 2024), [Online]. Available: <https://www.holisticai.com/blog/environmental-impact-ai-llms> (visited on 05/22/2024).
- [17] D. Patterson, J. Gonzalez, Q. Le, *et al.*, "Carbon emissions and large neural network training,"
- [18] P. Li, J. Yang, M. A. Islam, and S. Ren, *Making AI less "thirsty": Uncovering and addressing the secret water footprint of AI models*, Oct. 29, 2023. arXiv: 2304.03271[cs]. [Online]. Available: <http://arxiv.org/abs/2304.03271> (visited on 05/22/2024).
- [19] K. Foy. "AI models are devouring energy. tools to reduce consumption are here, if data centers will adopt. | MIT lincoln laboratory." (Sep. 22, 2023), [Online]. Available: <https://www.ll.mit.edu/news/ai-models-are-devouring-energy-tools-reduce-consumption-are-here-if-data-centers-will-adopt> (visited on 05/22/2024).
- [20] S. Gultekin, A. Globo, A. Zugarini, M. Ernandes, and L. Rigutini, "An energy-based comparative analysis of common approaches to text classification in the legal domain," in *AI, Machine Learning and Applications*, Academy & Industry Research Collaboration Center, Jan. 27, 2024, pp. 31–41, ISBN: 978-1-923107-17-5. DOI: 10.5121/csit.2024.140203. [Online]. Available: <https://aircconline.com/csit/papers/vol14/csit140203.pdf> (visited on 05/22/2024).
- [21] C. Morris. "The ethical pros and cons of meta's new llama 3 open-source AI model," Fast Company. (Apr. 20, 2024), [Online]. Available: <https://www.fastcompany.com/91109988/ethics-meta-llama-3-open-source-ai> (visited on 05/23/2024).
- [22] I. Hazewindus. "Open source large language models: An ethical reflection," *Filosofie in actie*. (), [Online]. Available: <https://www.filosofieinactie.nl/blog/2023/9/5/open-source-large-language-models-an-ethical-reflection> (visited on 05/23/2024).
- [23] C. Hashemi-Pour. "What is LangChain and how to use it: A guide," *Enterprise AI*. (Sep. 2024), [Online]. Available: <https://www.techtarget.com/searchenterpriseai/definition/LangChain> (visited on 05/23/2024).
- [24] LangChain. "Quick reference | LangChain." (), [Online]. Available: [https://python.langchain.com/v0.1/docs/modules/model\\_io/prompts/quick\\_start/](https://python.langchain.com/v0.1/docs/modules/model_io/prompts/quick_start/) (visited on 05/24/2024).
- [25] LangChain. "Custom chat model | LangChain." (), [Online]. Available: [https://python.langchain.com/v0.1/docs/modules/model\\_io/chat/custom\\_chat\\_model/](https://python.langchain.com/v0.1/docs/modules/model_io/chat/custom_chat_model/) (visited on 05/24/2024).

- 
- [26] LangCahin. “Output parsers | LangChain.” (), [Online]. Available: [https://python.langchain.com/v0.1/docs/modules/model\\_io/output\\_parsers/](https://python.langchain.com/v0.1/docs/modules/model_io/output_parsers/) (visited on 05/24/2024).
- [27] H. Sahota. “Implementing agents in LangChain,” Comet. (Nov. 30, 2023), [Online]. Available: <https://www.comet.com/site/blog/implementing-agents-in-langchain/> (visited on 05/23/2024).
- [28] RelevanceAI. “What is a multi agent system - relevance AI.” (), [Online]. Available: <https://relevanceai.com/learn/what-is-a-multi-agent-system> (visited on 05/23/2024).
- [29] Langsmith. “LangSmith.” (), [Online]. Available: <https://www.langchain.com/langsmith> (visited on 05/23/2024).
- [30] LangChain. “LangGraph | LangChain.” (), [Online]. Available: <https://python.langchain.com/v0.1/docs/langgraph/> (visited on 05/23/2024).
- [31] R. Merritt. “What is retrieval-augmented generation aka RAG?” NVIDIA Blog. (Nov. 15, 2023), [Online]. Available: <https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/> (visited on 05/23/2024).
- [32] R. Tripathi. “What are vector embeddings | pinecone.” (Jun. 30, 2023), [Online]. Available: <https://www.pinecone.io/learn/vector-embeddings/> (visited on 05/23/2024).
- [33] B. R. Swain. “Vector stores in LangChain,” Medium. (Apr. 9, 2024), [Online]. Available: <https://medium.com/@varsha.rainer/vector-stores-in-langchain-ca92b919cc3b> (visited on 05/23/2024).
- [34] M. Denscombe, *The Good Research Guide: For Small-scale Social Research Projects*. Open University Press, 2010, ISBN: 9780335213030.
- [35] S. B. Merriam, *Qualitative Research : A Guide to Design and Implementation*. Incorporated, Newark. Ebook Central: John Wiley & Sons, 2015.
- [36] S. Brinkmann and S. Kvale, *Doing Interviews* (Qualitative Research Kit), 2nd ed. London, England: SAGE Publications, Oct. 2018.
- [37] R. Patel, *Forskningsmetodikens grunder : Femte upplagan*. Lund : Studentlitteratur, [2019].
- [38] C. Qian, X. Cong, W. Liu, *et al.*, *Communicative agents for software development*, Dec. 19, 2023. DOI: 10.48550/arXiv.2307.07924. arXiv: 2307.07924[cs]. [Online]. Available: <http://arxiv.org/abs/2307.07924> (visited on 01/30/2024).
- [39] J. S. Park, J. C. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, *Generative agents: Interactive simulacra of human behavior*, Aug. 5, 2023. DOI: 10.48550/arXiv.2304.03442. arXiv: 2304.03442[cs]. [Online]. Available: <http://arxiv.org/abs/2304.03442> (visited on 01/30/2024).
- [40] G. Wang, Y. Xie, Y. Jiang, *et al.*, *Voyager: An open-ended embodied agent with large language models*, Oct. 19, 2023. DOI: 10.48550/arXiv.2305.16291. arXiv: 2305.16291[cs]. [Online]. Available: <http://arxiv.org/abs/2305.16291> (visited on 01/30/2024).
- [41] J. Saldaña, *The coding manual for qualitative researchers*, en, 2. ed. Los Angeles, Calif.: SAGE Publ, 2013, ISBN: 978-1-4462-4736-5 978-1-4462-4737-2.
- [42] V. Braun and V. Clarke, “Using thematic analysis in psychology,” en, *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, Jan. 2006. DOI:

- 10.1191/1478088706qp063oa. [Online]. Available: <http://dx.doi.org/10.1191/1478088706qp063oa>.
- [43] H. Snyder, "Literature review as a research methodology: An overview and guidelines," *Journal of Business Research*, vol. 104, pp. 333–339, Nov. 1, 2019, ISSN: 0148-2963. DOI: 10.1016/j.jbusres.2019.07.039. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0148296319304564> (visited on 04/26/2024).
- [44] K. T. Ulrich and S. D. Eppinger, *Product design and development*, en, Sixth edition. New York, NY: McGraw-Hill Education, 2016, ISBN: 978-0-07-802906-6.
- [45] *Manifesto for agile software development*, <https://agilemanifesto.org/>.
- [46] B. Ceylan, *Large language model evaluation in 2024: 5 methods*, en, <https://research.aimultiple.com/large-language-model-evaluation/>, Jan. 2024.
- [47] J. Wei, X. Wang, D. Schuurmans, *et al.*, "Chain of thought prompting elicits reasoning in large language models," *ArXiv*, Jan. 28, 2022. [Online]. Available: <https://www.semanticscholar.org/paper/1b6e810ce0afd0dd093f789d2b2742d047e316d5> (visited on 02/01/2024).
- [48] Z. Robertson, *GPT4 is slightly helpful for peer-review assistance: A pilot study*, Jun. 16, 2023. DOI: 10.48550/arXiv.2307.05492. arXiv: 2307.05492[cs]. [Online]. Available: <http://arxiv.org/abs/2307.05492> (visited on 03/19/2024).
- [49] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving factuality and reasoning in language models through multiagent debate," 2023, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2305.14325. [Online]. Available: <https://arxiv.org/abs/2305.14325> (visited on 02/01/2024).
- [50] Y. Talebirad and A. Nadiri, *Multi-Agent Collaboration: Harnessing the Power of Intelligent LLM Agents*, arXiv:2306.03314 [cs], Jun. 2023. [Online]. Available: <http://arxiv.org/abs/2306.03314> (visited on 02/14/2024).
- [51] S. M. Bsharat, A. Myrzakhan, and Z. Shen, *Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4*, en, arXiv:2312.16171 [cs], Jan. 2024. [Online]. Available: <http://arxiv.org/abs/2312.16171> (visited on 05/19/2024).
- [52] S. Yao, J. Zhao, D. Yu, *et al.*, "ReAct: Synergizing reasoning and acting in language models," *ArXiv*, Oct. 6, 2022. [Online]. Available: <https://www.semanticscholar.org/paper/99832586d55f540f603637e458a292406a0ed75d> (visited on 02/01/2024).
- [53] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, *Reflexion: Language agents with verbal reinforcement learning*, Oct. 10, 2023. DOI: 10.48550/arXiv.2303.11366. arXiv: 2303.11366[cs]. [Online]. Available: <http://arxiv.org/abs/2303.11366> (visited on 02/01/2024).
- [54] S. Min, X. Lyu, A. Holtzman, *et al.*, *Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?* Oct. 2022. DOI: 10.48550/arXiv.2202.12837. [Online]. Available: <http://arxiv.org/abs/2202.12837> (visited on 04/17/2024).

- [55] Y. Gao, Y. Xiong, X. Gao, *et al.*, *Retrieval-Augmented Generation for Large Language Models: A Survey*, arXiv:2312.10997 [cs], Jan. 2024. DOI: 10.48550/arXiv.2312.10997. [Online]. Available: <http://arxiv.org/abs/2312.10997> (visited on 02/08/2024).
- [56] Z. Xi, W. Chen, X. Guo, *et al.*, *The Rise and Potential of Large Language Model Based Agents: A Survey*, arXiv:2309.07864 [cs], Sep. 2023. [Online]. Available: <http://arxiv.org/abs/2309.07864> (visited on 02/01/2024).
- [57] W. Lu, J. Zhang, J. Zhang, and Y. Chen, *Large Language Model for Table Processing: A Survey*, arXiv:2402.05121 [cs] version: 1, Feb. 2024. DOI: 10.48550/arXiv.2402.05121. [Online]. Available: <http://arxiv.org/abs/2402.05121> (visited on 04/02/2024).
- [58] A. Madaan, N. Tandon, P. Gupta, *et al.*, *Self-Refine: Iterative Refinement with Self-Feedback*, arXiv:2303.17651 [cs], May 2023. DOI: 10.48550/arXiv.2303.17651. [Online]. Available: <http://arxiv.org/abs/2303.17651> (visited on 04/11/2024).
- [59] *Pricing*, en-US. [Online]. Available: <https://openai.com/api/pricing/> (visited on 05/24/2024).
- [60] A. Papikyan. “Fine-tuning large language models (LLMs) in 2024 | SuperAnnotate.” en. (), [Online]. Available: <https://www.superannotate.com/blog/llm-fine-tuning> (visited on 05/24/2024).
- [61] *How to Install and Deploy LLaMA 3 Into Production?* en, Apr. 2024. [Online]. Available: <https://nlpcloud.com/how-to-install-and-deploy-llama-3-into-production.html> (visited on 04/27/2024).



# A

## Data collection

### A.1 English interview guide

#### Introduction of interviewers and purpose

1. Hi, we are Daniel and Felix and we are here at GKN Aerospace for our master's thesis project. In our thesis, we are looking at the analysis report generation process for product verification. From our understanding, the process is not very efficient, and it takes a lot of time and iterations for the reports to be approved. (Show pic eventually) The purpose of this interview is to find out more about how the process works, the inherent problems and how it can be improved.
2. Your answers will greatly help us in our thesis and will contribute to potentially improving the overall analysis report generation process.
3. If it is okay, we will record the interview. The only information that will be stored is your role and experience, otherwise the interview is anonymous. The recordings will be removed after they have been analyzed. The conclusions will be permanently stored and presented in the thesis with the exception of confidential content.

#### Introduction of interview subject:

1. Can you introduce yourself?
  - And tell us about your role and your responsibilities in the process?
  - How experienced with the process would you consider yourself to be?

#### Background and workflow information:

1. Describe the current workflow for analysis report generation (For your role):
  - What are the key-steps?
  - What tools and software are used?

#### Challenges and problems

1. What is the biggest challenge or problem related to the speed and efficiency of the process?
  - Are there any bottlenecks?
2. What do you think about communication and collaboration with the other roles in the process?

#### Improvements

1. How do you think the process can be improved?
2. What is your experience with AI tools such as ChatGPT in general?
3. *Potential discussion on different LLM-based frameworks and ideas*
4. What is your perception of the capabilities of such tools to aid in the analysis report generation process?

#### Closing:

1. Is there anything you would like to add?
2. Thank you "name" for taking your time and giving us your valuable insight. We appreciate it alot.
3. We will summarize the interview and get back to you on email of the important key-findings.
4. If there is anything you would like to add, feel free to contact us by replying to that mail.
5. Do you know any additional person that we can interview? (Writer or approver)



# B

## Concept development documents

### B.1 Dummy Report

**Summary:** Includes a summarizing table of the most important details of the analysis and the reached results.

**Nomenclature and Symbols:** Includes a list of the acronyms and symbols with the corresponding descriptions used throughout the report.

**Introduction:** Includes a background to the problem and description with an optional figure of the part to be analyzed. This section also includes the reasons and purpose of the analysis.

**References:** Includes the references to the documents used to perform the analysis.

**Requirements:** Includes the required criteria and limits that the analysis takes into account and relates to.

**Geometry:** Includes illustrative cad model images of the analyzed geometry along with an overall description of the model.

**Analysis model:** Includes reference to the finite element (FE) method used and illustrative figures for the FE model with its eventual submodels. Also includes options and settings for the FE analysis.

**Materials:** Includes reference to the material design record document used together with an explanatory description of the materials used in the analysis with their corresponding properties.

**Loads and Boundary Conditions:** Includes reference to the loads and loads transformation documents used in the analysis. The section also discusses the types of loads applied in the analysis together with explanations and constraints considered.

**Method:** Includes the methodology used to perform the analysis with the tools and software used. Also includes a reference to Design Practice (DP) for the selected methodology.

**Results:** Includes the results of the analysis carried out, with figures where appropriate, but also with tables and clear explanations about the findings.

**Conclusion:** Includes a conclusion of the analysis with a summary, if the technical requirements are satisfied or not, also with suggestions for future potential tests with argumentation and discussion of the results.

**File management:** Includes the location of long-term references along with all files to be saved. Also includes scripts, log files and e.g. Ansys files.

## B.2 Language guidelines

<p><b>Guideline, tense 1:</b></p> <p><i>Example:</i></p>	<p>Actions performed while writing the report must use present simple verb tense.</p> <p><i>When comparing and summarizing results created during the report, a good sentence is: The critical location has a stress of 345 MPa.</i></p>
<p><b>Guideline, tense 2:</b></p> <p><i>Example:</i></p>	<p>Actions performed during the analysis must use present perfect verb tense.</p> <p><i>Correct sentence: The FE model has been run with the following options.</i></p>
<p><b>Guideline, tense 3:</b></p> <p><i>Example:</i></p>	<p>Use past simple as the verb tense when referring to a previous report.</p> <p><i>Example of a correct sentence: The FE model on report [3] was run with the following options.</i></p>
<p><b>Guideline, passive voice 1:</b></p> <p><i>Example:</i></p>	<p>Passive voice must be used to remove the subjects I, You, She and We.</p> <p><i>Incorrect sentence: I calculate the results following the method described in the previous section.</i></p> <p><i>Correct sentence: The results are calculated following the method described in the previous section.</i></p>
<p><b>Guideline, passive voice 2:</b></p> <p><i>Example:</i></p>	<p>While using the passive voice an article should be included (the, a, an) if the subject is countable.</p> <p><i>Incorrect sentence: Results are calculated.</i></p> <p><i>Correct sentence: The results are calculated.</i></p>
<p><b>Guideline, personal names:</b></p> <p><i>Example:</i></p>	<p>No personal names.</p> <p><i>Incorrect sentence: Felix and Daniel amended the loads due to...</i></p> <p><i>Correct sentence: Loads were amended due to...</i></p>
<p><b>Guideline, customer's names:</b></p> <p><i>Example:</i></p>	<p>Avoid mentioning GKN or the customer's name such as "Orion Aerospace Propulsion" and "Galactic Drive Innovation" in the report.</p> <p><i>Incorrect sentence: New loads received by GKN have increased when compared to previous loads sent by Orion Aerospace Propulsion.</i></p> <p><i>Correct sentence: Loads received have increased when compared to previous loading.</i></p>

**Table B.1:** Guidelines for language with a corresponding example for each guideline.

### B.3 Content guidelines

<b>Introduction:</b>	<ol style="list-style-type: none"> <li>1. The background and purpose of the report</li> <li>2. The type of analysis performed</li> <li>3. A figure of the component</li> </ol>
<b>References:</b>	<ol style="list-style-type: none"> <li>1. The section should contain ONLY a list of references.</li> <li>2. The list of references should be numbered in square brackets: [1], [2], ...</li> <li>3. Each reference should contain the following information: <ul style="list-style-type: none"> <li>- The report number, in the format DBXXXXXX, VOLS:XXXXXXXX or AP:XXXXXX</li> <li>- A "-" separator</li> <li>- The title of the report.</li> </ul> </li> </ol>
<b>Requirements:</b>	<ol style="list-style-type: none"> <li>1. The requirement section shall describe all the requirements that are going to be checked against in the report.</li> <li>2. The source of the requirement shall be clearly stated and a reference given in brackets: [1], [2],...</li> <li>3. Besides the points above, no other content shall be allowed in this section.</li> </ol>
<b>Geometry:</b>	<ol style="list-style-type: none"> <li>1. A CAD model figure caption, including dummy surrounding components.</li> <li>2. An explicit reference of a CAD model (unique reference via a ref ID)</li> <li>3. A sentence explaining if the model is based on nominal or minimum dimensions.</li> </ol>
<b>Analysis models:</b>	<ol style="list-style-type: none"> <li>1. A Reference to the Finite Element model report.</li> <li>2. A Figure caption showing the mesh of the FE model.</li> </ol>
<b>Material:</b>	<ol style="list-style-type: none"> <li>1. A reference to the material design record document.</li> <li>2. A figure caption of the TRS and surrounding dummy component.</li> <li>3. A description or a table of the materials that the component and surrounding parts are made of.</li> <li>4. If there are finite element sub-models, there should be a new table or description of the materials and a corresponding picture.</li> <li>5. An explicit mention on the material properties type of data used: minimum or average. (or if there are any factors that will be applied to take it into account).</li> </ol>

	<p>6. An explicit mention of the material properties that will be used in the analysis: Ultimate strength or Yield strength.</p>
<p><b>Loads and Boundary Conditions:</b></p>	<p>1. A subsection called Loads, containing:</p> <ul style="list-style-type: none"> <li>- What Mechanical Loads are applied in the section (a reference to the document).</li> <li>- What Thermal Loads are applied in the section (a reference to the document).</li> <li>- Description of how pressure Loads are applied in the section (a reference to the document).</li> <li>- A description of how the contacts are applied.</li> <li>- A description of the pilot nodes where the loads will be applied.</li> <li>- A description of how the Multi Point Constraints (MPC) are applied and where.</li> </ul> <p>2. An optional subsection called ‘load cases’, containing:</p> <ul style="list-style-type: none"> <li>- A description of the number of load cases applied on the linear run.</li> <li>- An explanation if all the load cases provided have been run, or if there has been a down-selection process and how this process was performed.</li> <li>- What load cases have been selected for non-linear analysis and why.</li> <li>- A table describing the original load case number (as provided by the loads document or customer) and the load case number of this report.</li> </ul>
<p><b>Method:</b></p>	<p>1. An explicit reference to the Design Practice (DP) and Design Verification Plan (DVP) that drives the analysis method in this report.</p> <p>2. The method section shall not use older analysis reports to justify how the analysis shall be conducted.</p> <p>3. A subsection called ‘Tools’ that contain:</p> <ul style="list-style-type: none"> <li>- A list of in house tools used.</li> <li>- Each tool in the list shall have a reference document.</li> <li>- the version of each tool shall be specified.</li> </ul> <p>4. A subsection called ‘Software’ that contain:</p> <ul style="list-style-type: none"> <li>- The Version of Ansys used.</li> </ul> <p>5. A subsection called ‘Method Procedure’ containing:</p> <ul style="list-style-type: none"> <li>- If the method used in the analysis deviates from the Design Practice (DP) or Design Verification Plan (DVP).</li> </ul>

	<ul style="list-style-type: none"> <li>- Additional calculations that have not been specified in the Design Practice (DP) or Design Verification Plan (DVP).</li> <li>6. An optional subsection to describe how the method has been applied, step by step.</li> </ul>
<b>Results:</b>	<ol style="list-style-type: none"> <li>1. This section shall contain only the stress results, in particular: <ul style="list-style-type: none"> <li>- Do not include any information about the process, that shall be included in the method section.</li> <li>- Do not include any information about load factors, that shall be included in the Loads section.</li> <li>- Do not include any information about material properties or knock-down factors, that shall be included in the Loads section.</li> </ul> </li> <li>2. The results shall be well presented, preferably in a summarized table.</li> </ol>
<b>Conclusions:</b>	<ol style="list-style-type: none"> <li>1. A statement about if the technical requirements are satisfied or not.</li> <li>2. Conclusion of the analysis.</li> <li>3. Proposal or suggestions for the future if needed</li> </ol>

**Table B.2:** Guidelines for content for each section of the report.

# C

## Test cases and results

### C.1 Design criteria: Review language

Test case 1: Language				
No.	Type:	Original:	Error description:	Location:
1	Customer name	The linear static analysis has been performed by applying the limit loads at the various flange interfaces and the mounts.	The linear static analysis has been performed by applying the limit loads <b>given by Galactic Drive Innovation</b> at the various flange interfaces and the mounts.	1 Introduction
2	Personal name	In the event of criteria 2a not being fulfilled at some locations, or the values too close to the limit according to engineering judgement, ...	In the event of criteria 2a not being fulfilled at some locations, or the values too close to the limit according to <b>Najeems</b> judgement,	3.3 Criteria 2b
3	Tense	For regular panels, this has been taken into account by using overaged Material A sheet material properties in the non-linear analyses and also linear analysis margins have been evaluated using overaged allowable yield strengths (see section 6.2).	For regular panels, this <b>will be</b> taken into account by using overaged Material A sheet material properties in the non-linear analyses and also linear analysis margins <b>will be</b> evaluated using overaged allowable yield strengths (see section 6.2).	7.1 Loads
4	Customer name	Pressure loads have been small in comparison to limit loads and therefore have been neglected;	Pressure loads have been small in comparison to limit loads and therefore have been neglected <b>according to Orion Aerospace Propulsion</b> ;	7.1 Loads
5	Passive voice	it is suggested to consider the learnings from this report,	<b>we suggest</b> to consider the learnings from this report,	8.2 Method procedure
6	Passive voice	A non-linear analysis has been carried out for the DLCs which has been resulting in negative linear analysis margins (i.e. where some plasticity is found).	<b>I have been carrying</b> out a non-linear analysis for the DLCs which has been resulting in negative linear analysis margins (i.e. where some plasticity is found).	8.2.2 Non-Linear Analysis for Strain Margins
7	Passive voice	The Python script "get_plastic_strain.py" has been used for this procedure.	<b>The</b> Python script "get_plastic_strain.py" has been used for this procedure.	8.2.5 Plastic strain estimation
8	Tense	A non-linear analysis has thus been carried out for the critical load case only.	A non-linear analysis <b>will thus be</b> carried out for the critical load case only.	9.2 Non-linear analysis
9	Personal name	The estimated plastic strain from the interpolation of stress strain at 900K for Material A for the obtained von Mises stress of 601 MPa (see Figure 9 5) is found to be 0.21%	The estimated plastic strain from the interpolation of stress strain at 900K for Material A for the obtained von Mises stress of 601 MPa (see Figure 9 5) is found <b>by Alex</b> to be 0.21%	9.2.2 Sub-model analysis
10	Passive voice	However, because the residual stress from the sub-model run (300 MPa) is significantly higher than that of the 360 FE model run (60 MPa),	However, because the residual stress from <b>the</b> sub-model run (300 MPa) is significantly higher than that of <b>the</b> 360 FE model run (60 MPa),	10 Conclusion
11	Tense	All other components show positive margin and therefore fulfill the DP [4] requirement criterion 1 with the equivalent von Mises stress less than average yield strength.	All other components <b>will show</b> positive margin and therefore <b>will fulfill</b> the DP [4] requirement criterion 1 with the equivalent von Mises stress less than average yield strength.	10 Conclusion

Table C.1: The test case used to evaluate DC1

C. Test cases and results

Err.No	Language 1			Language 2			Language 3		
	Mistral 8x7b		GPT 4	Mistral 8x7b		GPT 4	Mistral 8x7b		GPT 4
	Run 1	Run 2	Run 1	Run 1	Run 2	Run 1	Run 1	Run 2	Run 1
	<i>Spotted errors</i>			<i>Spotted errors</i>			<i>Spotted errors</i>		
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
	<b>Percentage intentional errors found</b>			<b>Percentage intentional errors found</b>			<b>Percentage intentional errors found</b>		
	0%	0%	45%	36%	45%	55%	55%	55%	73%
	<b>Not accurate errors</b>			<b>Not accurate errors</b>			<b>Not accurate errors</b>		
	78	76	3	131	133	62	166	143	73
	<b>Percentage accurate feedback</b>			<b>Percentage accurate feedback</b>			<b>Percentage accurate feedback</b>		
	0%	0%	63%	3%	4%	9%	3%	4%	10%
	<b>Tokens</b>			<b>Tokens</b>			<b>Tokens</b>		
	13,061	12,965	5,822	60,528	63,676	48,873	91,197	88,632	83,504
	<b>Seconds</b>			<b>Seconds</b>			<b>Seconds</b>		
	90	90	33	27	35	41	19	23	48

Table C.2: The results from the tests to evaluate DC1

## C.2 Design criteria: Review content

Test case 2: Content				
No.	Type:	Error sentence:	Error description:	Location:
1	Missing		No background & purpose	1. Introduction
2	Misplaced	"Where the linear margins have been found to be negative, a non-linear analysis has been carried out for the corresponding load cases. The method of compliance was given by the Verification plan [3]. The research method and analysis resulted in concerning stresses along the stator blades."	Mention of some results in the introduction	1. Introduction
3	Misplaced	"[1] Wikipedia – Ducks life in the world"	A totally unrelated reference	2. References
4	Misplaced	<b>1.1 Criteria 3a</b>	A section that talks about innebandy on fridays	3. Requirements
5	Missing		No reference to requirement source	3. Requirements
6	Missing		No mention of nominal or minimal dimensions	4. Geometry
7	Missing		No reference to FE model report	5. Analysis model
8	Misplaced	"However, an analysis using Ansys was performed by applying loads on the critical locations which resulted in relatively high stress factors."	Mentioning analysis and results in the loads section	7. Loads and Load boundaries
9	Missing		Section "tools" missing	8. Method
10	Misplaced	"As explained in method section 8.2.4, the maximum allowed material knockdown for the Material B struts, to sustain a positive elastic strength margin according to DP [4] criterion 1, is calculated as shown below in Table 9 5."	Describing the process in the results section	9. Results
11	Misplaced	"After removing the duplicates, 23 unique limit load cases are identified for limit analysis on the 360 FE model. The dimensioning load case (DLC) and Margin to yield against average yield strength are obtained and shown in Table 9 1"	Describing the process in the results section	9. Results
12	Misplaced	"As described in the section 8.2.6, the cut-boundary displacements and nodal temperatures have been applied to the sub-model."	Describing the process in the results section	9. Results
13	Misplaced	"A non-linear analysis has thus been carried out"	Describing the process in the results section	9. Results
14	Misplaced	"An elastic-plastic analysis has been performed to determine the redistributed stress and strain at the critical location with applied load factor of 1.15. The Non-Linear analysis has only been performed for the critical load case for which the negative margin has been observed in the linear analysis. "	Describing the process in the results section	9. Results
15	Missing		No statement on if requirement is met	10. Conclusion

Table C.3: The test case used to evaluate DC2

Err.No	Content 1			Content 2 (Reflection)		
	Mistral 8x7b		GPT 4	Mistral 8x7b		GPT 4
	Run 1	Run 2	Run 1	Run 1	Run 2	Run 1
	<i>Spotted errors</i>			<i>Spotted errors</i>		
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
	<b>Percentage intentional errors found</b>			<b>Percentage intentional errors found</b>		
	47%	53%	73%	53%	60%	67%
	<b>Not accurate errors</b>			<b>Not accurate errors</b>		
	13	12	11	20	16	7
	<b>Percentage accurate feedback</b>			<b>Percentage accurate feedback</b>		
	35%	40%	50%	29%	36%	59%
	<b>Tokens</b>			<b>Tokens</b>		
	11 494	11 670	9 910	41 253	40 596	37 154
	<b>Seconds</b>			<b>Seconds</b>		
	3	5	16	7	5	24

Table C.4: The results from the tests to evaluate DC2

## C.3 Design criteria: Review correct reference usage

Test case 3: Correct reference					
No.	Type:	Original sentence:	Error:	Comment:	Location and ref.
1	Wrong reference			The FE-model report is actually a description about lego construction	5.1, 5.2, 6.1 and [1]
2	Wrong information	The limit analysis has been carried out at redline thermal point [11].	The limit analysis has been carried out at <b>yellowline</b> thermal point [11].	Reference mentions nothing about yellow line thermal point	7.1.2 Thermal loads and [11]
3	Wrong information	Redline temperature field [11] has been applied throughout the analysis with thermal expansion and the reference temperature has been set as 298.4K (see section 7.1.2).	<b>The magnetic</b> field [11] has been applied throughout the analysis with thermal expansion and the reference temperature has been set as 298.4K (see section 7.1.2).	Reference [11] mentions nothing about a magnetic field	7.3 Constraints and [11]
4	Wrong information	The maximum von Mises stress calculated by the FE model at 115% loading conditions shall be lower than the yield strength of the material (average properties)	The maximum von Mises stress calculated by the FE model at 115% loading conditions shall be lower than the yield strength of the material ( <b>minimal</b> properties)	The fake Design Practice states that average material properties are used	3.1 Criteria 1 and [4]
5	Wrong information	The plastic strain at the critical location must be less than or equal to 0.20% per criterion 2a [4]. If criterion 2a is not fulfilled, criterion 2b is sought.	The plastic strain at the critical location must be less than or equal to <b>0.15%</b> per criterion 2a [4]. If criterion 2a is not fulfilled, criterion 2b is sought.	Fake Design Practice states it should be 0.20%	8.2.2 Non-Linear Analysis for Strain Margins and [4]
6	Wrong information	Both 360 and sub-model non-linear analysis results are summarized in Table 9-3. The max remaining deformation of 123 $\mu\text{m}$ is less than 50% of the drawing tolerance. The limit requirement is thus fulfilled with respect to criterion 2b of DP [4].	Both 360 and sub-model non-linear analysis results are summarized in Table 9-3. The max remaining deformation of 123 $\mu\text{m}$ is less than <b>55%</b> of the drawing tolerance. The limit requirement is thus fulfilled with respect to criterion 2b of DP [4].	Fake DP states that it should be less than half	9.2 Non-linear analysis and [4]
7	Wrong information	In the event of criteria 1 not being fulfilled at some locations, the criteria 1 can be substituted at those locations with this criteria: The plastic von Mises strain shall be lower than 0.2% (0.002) at 115% mechanical loading.	In the event of criteria 1 not being fulfilled at some locations, the criteria 1 can be substituted at those locations with this criteria: The plastic von Mises strain shall be lower than 0.2% (0.002) at <b>120%</b> mechanical loading.	Fake Design Practice states 115 %.	3.2 Criteria 2a and [4]
8	Wrong information	per previous analyses [13] and Design Practice [4].	per previous analyses [13] <b>that talks about the thermal redline strength within aircraft wings</b> and Design Practice [4].	[13] does not mention any of the text in red	7.1 Loads and [13]
9	Wrong information	There are 86 Limit load cases (LC1 to LC86) provided in the Design Loads Document [10]	There are <b>90</b> Limit load cases (LC1 to <b>LC90</b> ) provided in the Design Loads Document [10]	The report states 80. The reference [10] itself states that there are 86 limit load cases.	7.1.1 Mechanical loads and [10]
10	References not in folder			There are 10 references used without the reference in the actual folder	
11	Wrong reference used	As explained in method section 8.2.4, the maximum allowed material knockdown for the Material B struts, to sustain a positive elastic strength margin according to DP [4] criterion 1, is calculated as shown below in Table 9-5	As explained in method section 8.2.4, the maximum allowed material knockdown for the Material B struts, to sustain a positive elastic strength margin according to DP <b>[11]</b> criterion 1, is calculated as shown below in Table 9-5	Reference [11] is used instead of [4]	9.3 Overaging in struts, [11] and [4]
12	Wrong reference used	The Limit analysis for the 360-degree solid model has been performed according to DP [4] and DVP [3]	The Limit analysis for the 360-degree solid model has been performed according to DP [4] and DVP <b>[2]</b>	Reference [2] is used instead of [3]	8.2 Method procedure, [2] and [3]
13	Wrong reference used	The strength requirements of the TRS subjected to Limit loads was given by the Technical requirement specification [2] as shown in Table 3 1. Table 3 1 Limit Strength requirements from the Technical Requirement Specification [2]	The strength requirements of the TRS subjected to Limit loads was given by the Technical requirement specification <b>[11]</b> as shown in Table 3 1. Table 3 1 Limit Strength requirements from the Technical Requirement Specification <b>[11]</b>	Reference [11] is used instead of [2]	2. Requirements, [11] and [2]

Table C.5: The test case used to evaluate DC3

	Correct ref 1 (section)			Correct ref 2 (chunk)		
	GPT 4			GPT 4		
	Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
Err.No	<i>Spotted errors</i>			<i>Spotted errors</i>		
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
	<b>Percentage intentional errors found</b>			<b>Percentage intentional errors found</b>		
	46%	38%	54%	54%	69%	62%
	<b>Not accurate errors</b>			<b>Not accurate errors</b>		
	4	2	2	10	10	12
	<b>Percentage accurate feedback</b>			<b>Percentage accurate feedback</b>		
	60%	71%	78%	41%	47%	40%
	<b>Tokens</b>			<b>Tokens</b>		
	15 060	15 002	15 073	18 253	19 320	19 350
	<b>Seconds</b>			<b>Seconds</b>		
	205	210	205	280	305	320

Table C.6: The results from the tests to evaluate DC3

## C.4 Design criteria: Review consistency

Test case 4: Consistency				
No.	Original:	Error sentence:	Comment:	Location:
1	Criterion 2b prescribes a non-linear analysis considering loading to 115% mechanical load and unload to 0% load and room temperature, to study the residual plastic strain and total displacement	Criterion 2b prescribes a non-linear analysis considering loading to 110% mechanical load and unload to 0% load and room temperature, to study the residual plastic strain and total displacement	Criteria 2b considers 115% loading	8.2.3 Non-linear analysis
2	Linear static analysis of the 360 model has been carried out with 115% mechanical (limit) loads and redline thermals [11], and the DLC has been identified for the various regions of the TRS per the Design Practice [4].	Linear thermal analysis of the 360 model has been carried out with 115% mechanical (limit) loads and redline thermals [11], and the DLC has been identified for the various regions of the TRS per the Design Practice [4].		8.2.1 Linear analysis
3	Loads have been applied using multipoint constraint (MPC) algorithm.	The environmental loads have been applied using multipoint constraint (MPC) algorithm.		7.1.1 Mechanical loads
4	Redline temperature field [11] has been applied throughout the analysis with thermal expansion and the reference temperature has been set as 294.8K (see section 7.1.2)	Redline temperature field [11] has been applied throughout the analysis with thermal expansion and the reference temperature has been set as 298.4K (see section 7.1.2)	Says 294.8 across report	7.3 Constraints
5	However, because the residual stress from the sub-model run (318 MPa)	However, because the residual stress from the sub-model run (300 MPa)	In the results section it is 318 Mpa	10 Conclusion
6	For the regular panels, overaged Material A properties have been used to consider the effect of thermal degradation of material in the non-linear analysis.	For the regular panels, Material A properties have been used to consider the effect of thermal degradation of material in the non-linear analysis.	Method section says overaged was used	10 Conclusion
7	This means that, when overaged data becomes available for Material B, should the actual knockdown stress prove to be greater than 200 MPa, the struts fail to meet criterion 1 of the limit analysis DP [4] and further non-linear analysis is necessary for validation of the struts.	This means that, when overaged data becomes available for Material B, should the actual knockdown stress prove to be greater than 300 MPa, the struts fail to meet criterion 1 of the limit analysis DP [4] and further non-linear analysis is necessary for validation of the struts.	200 MPa used elsewhere	10 Conclusion
8	Average material properties have been used in the analysis.	Minimum material properties have been used in the analysis.	Average material properties are used	10 Conclusion
9	For linear analyses the margin of safety, with respect to yield strength for the identified components under critical load, has been calculated using the formula shown below	For linear analyses the margin of safety, with respect to ultimate strength for the identified components under critical load, has been calculated using the formula shown below	Yield strength should be used	8.2.1 Linear analysis
10	The max remaining deformation of 1.23 $\mu\text{m}$ is less than 50% of the drawing tolerance.	The max remaining deformation of 123 $\mu\text{m}$ is less than 50% of the drawing tolerance.	Should be 1.23	9.2 Non-linear analysis
11	The Von Mises stress and plastic strain plots at 115% load are shown in Figure 9-5 and Figure 9-6.	The Carl von Linné stress and plastic strain plots at 115% load are shown in Figure 9-5 and Figure 9-6.	von Mises in rest of report	9.2.2 Sub-model analysis
12	After removing the duplicates, 23 unique limit load cases are identified for limit analysis on the 360 FE model.	After removing the duplicates, 25 unique limit load cases are identified for limit analysis on the 360 FE model.	Loads section says that there are 23 unique limit load cases	9.1 Linear analysis

Table C.7: The test case used to evaluate DC4

<b>Consistency 1</b>			
<b>GPT 4</b>			
	<b>Run 1</b>	<b>Run 2</b>	<b>Run 3</b>
<b>Err.No</b>	<i>Spotted errors</i>		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
<b>Percentage intentional errors found</b>			
	67%	58%	58%
<b>Not accurate errors</b>			
	8	9	10
<b>Percentage accurate feedback</b>			
	50%	44%	41%
<b>Tokens</b>			
	38 998	33 772	34 048
<b>Seconds</b>			
	305	291	376

**Table C.8:** The results from the tests to evaluate DC4

## C.5 Design criteria: Review missing references

Test case 5: Missing references				
No.	Type:	Original:	Error description:	Location:
1	Removed ref	The strength requirements of the TRS subjected to Limit loads was given by the Technical requirement specification [2] as shown in Table 3 1.	The strength requirements of the TRS subjected to Limit loads was given by the Technical requirement specification [2] as shown in Table 3 1.	3 Requirements
2	Removed refs	See also drawings.	See also drawings [7] and [8].	4 Geometry
3	Removed refs	The details of FE model is found in the FE model description report.	The details of FE model is found in the FE model description report [1].	5.1 360 model
4	Removed fig ref	This sub-model, figure, includes surface shell elements over the entire model.	This sub-model, figure 5-3, includes surface shell elements over the entire model.	5.2
5	Removed refs	The Limit analysis for the 360-degree solid model has been performed according to DP and DVP.	The Limit analysis for the 360-degree solid model has been performed according to DP [4] and DVP [3].	8.2 Method procedure

**Table C.9:** The test case used to evaluate DC5

	Missing ref 1 (Full)			Missing ref 2 (chunk)		
	GPT 4			GPT 4		
	Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
Err.No	<i>Spotted errors</i>			<i>Spotted errors</i>		
1						
2						
3						
4						
5						
	<b>Percentage intentional errors found</b>			<b>Percentage intentional errors found</b>		
	40%	40%	40%	80%	80%	80%
	<b>Not accurate errors</b>			<b>Not accurate errors</b>		
	3	4	3	14	12	13
	<b>Percentage accurate feedback</b>			<b>Percentage accurate feedback</b>		
	40%	33%	40%	22%	25%	24%
	<b>Tokens</b>			<b>Tokens</b>		
	5 098	5 297	5 216	10 317	9 881	9 913
	20	33	25	27	23	23

Table C.10: The results from the tests to evaluate DC5

Department of Industrial and Materials Science  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden  
[www.chalmers.se](http://www.chalmers.se)



**CHALMERS**