EVALUATING LARGE LANGUAGE MODELS FOR SPACE OPERATIONS

Clemens Schefels*, Carsten Hartmann*, Kathrin Helmsauer*, Leonard Schlag*
* German Aerospace Center (DLR), German Space Operations Center (GSOC), Weßling, Bavaria, 82234 Germany

Abstract

The increasing number of orbital launches has led to a significant surge in spacecrafts in Earth's orbit, making mission control centers more complex and challenging to manage. To address this issue, we investigated the potential of Large Language Models (LLMs) in space operations, focusing on tasks such as information retrieval, documentation, and quick decision-making. This paper explores two evaluation studies that tested the application of LLMs for specific use cases in a non-cloud environment due to sensitive data confidentiality requirements, including a scenario for the Columbus Flight Control Team to quickly retrieve information during operations. Our results demonstrate the promising capabilities of LLMs in supporting spacecraft engineers with answers, while also highlighting the need for parameter fine-tuning, prompt engineering, or even model re-training. This paper provides actionable insights into the potential integration of LLMs in space operations and outlines future research directions related to this emerging field.

Keywords

Large Language Models; Space Operations; Retrieve Information

1. INTRODUCTION

Each year, the number of orbital launches increases dramatically – for example, in 2023, 211 orbital launches were carried out worldwide, 33 more than the year before and 129 more than ten years before. With the growing number of launches, the number of spacecrafts in the earth's orbit increases as well. As a result, mission control centers nowadays need to manage a greater number of spacecrafts as well as the increasing workload and complexity. Therefore, new technologies for reducing the workload of their employees while maintaining safe, reliable, and economic space operations need to be adapted.

One new technology that already revolutionizes the daily work routine in many domains are foundation models: AI-models that are trained on huge amounts of data. These models have shown astonishing capabilities in different fields and applications; from generating photorealistic images or writing human like texts to producing true-to-life video sequences. Especially Large Language Models (LLMs), which comprehend natural language in an unprecedented way, are a promising new technology which may be able to enhance space operations and help reduce the steadily increasing workload.

The German Space Operations Center (GSOC) at the German Aerospace Center (DLR) explored several challenging tasks at which spacecraft engineers could be supported by LLMs which will be described in this article. For example, engineers need to react quickly to issues and have to memorize and recall a large amount of information acquired from spacecraft documentations, flight procedures, etc. – all of which are textual data. In addition, engineers have a high documentation effort.

With LLM based tools, we could support the engineers both during training and operations by providing them with quick and reliable answers.

Since the data of mission control centers are highly confidential, the sensitive data cannot leave internal networks. Therefore, a non-cloud-solution is often needed in our domain. This in itself is a challenging task, since most of the popular LLM off-the-shelf products are cloud-based. In this article, we describe possible areas of application as well as two evaluation studies in which we tested the application of LLMs for the previously described tasks. In the first study, we investigated the basic features of LLMs and analyzed whether this new technology is suitable for space operations. In a follow-up study, we researched a concrete scenario for LLMs to provide the Columbus Flight Control Team with a tool to quickly retrieve information during operations. For each study, we provide the approach, the conclusions, and our lessons learned.

This paper is structured as follows: In the Motivation, we begin by providing a concise motivation for the need of an assistant system for space operations, highlighting the complexities and challenges. Related work and technological background, a broader understanding of the technological context is then offered, including relevant projects that have explored similar ideas. In section Use Case Experiments, the core contribution of this paper lies in its systematic and effectiv investigation of the capabilities of LLMs in space operation, exploring their potential to augment human decision-making and enhance operational efficiency. Our use cases are carefully designed to explore the strengths and limitations of LLMs in a short time. Finally, in Conclusion and Outlook, we conclude by distilling our findings into actionable insights and highlight future research directions related to the integration of LLMs in space operations.

2. MOTIVATION

With a permanently growing satellite fleet, operators and system engineers face the challenge of memorizing and recalling vast amounts of textual data from spacecraft documentation, flight procedures, etc. This information is crucial during training/operations, but the danger is high that in emergency situations, operators may struggle to recall the right information or spend too much time searching for it.

And, as the number of satellites increases, this leads to an increasing effort required for documentation too.

To address these issues, we need an assistance system that can search, collect, and summarize necessary information while supporting operators in documenting their work. Therefore, we've explored the capabilities of foundation models to support operators with these tasks through two experiments. The design aimed to provide quick results and demonstrate the potential for this technology. Our objective is to empower engineers with rapid, accurate, and reliable answers during both training and operational phases.
However, sensitive data must remain within our internal network, which restricts the use of cloud solutions since we neither own a private cloud nor have the capacities to implement and operate one.

3. TECHNICAL FOUNDATION AND RELATED WORK

This section provides an overview to the technical aspects of our experiments, with a focus on foundation models, i.e., large language models. Due to the breadth and depth of research in this area, we will only provide a brief overview, rather than going into detailed explanations that could fill several papers. We detail the specific methodologies, tools, and techniques employed in our experiments, which informed the design of our experimental setup.

3.1. LLMs in Short

The concept of Large Language Models (LLMs) was first introduced in the paper "Attention Is All You Need" by A. Vaswani et al. [1], which presented the Transformer network architecture as a novel approach to processing sequential data, such as text. This architecture involves converting text into numerical representations called tokens, which are then transformed into vectors through a word embedding table (a lookup table that maps words to dense vector representations). Each token is contextualized within a context window via a parallel multi-head attention mechanism, allowing the model to selectively amplify or diminish the importance of key tokens. This architecture has revolutionized Natural Language Processing (NLP), achieving impressive results and spawning numerous applications, including call center chatbots with human-like conversations and text generation. However, due to the resource-intensive nature of training LLMs on large datasets, only major tech companies like OpenAI, Google, and Meta have been able to train these models. To democratize access to LLMs and break the dominance of OpenAI's market leader, Meta and Google have published their models open-source, allowing a vast user community to develop around them. The Hugging Face platform [2] is a leading example of this ecosystem, offering a library of pre-trained models and tools for research and innovation.
To address concerns about data privacy within this community, frameworks have been developed to run LLMs locally on users' own hardware without requiring an internet connection. Two popular examples are private-gpt, which enables users to query their documents using the power of LLMs while maintaining complete control over their data, and LocalGPT, an open-source initiative that allows conversational interactions with documents without compromising user privacy. By running these models locally, users can be assured that no sensitive data leaves their computer, meeting our requirement for preserving confidentiality within our network.
To conclude this short introduction, we want to use the power of Large Language Models to provide an assistant to our operator and system engineers that can answer complex queries, help with troubleshooting, and facilitate knowledge access across our organization.

4. USE CASE EXPERIMENTS

To explore the potential of LLMs in supporting spacecraft operators and system engineers with their tasks, we conducted two experiments designed to yield quick results. Our primary objective is to enable engineers during training and operations by providing them with rapid and reliable answers.
However, our operational requirements present certain limitations: Due to security concerns, sensitive data must remain within our internal network. This restricts the use of cloud solutions, as we do not own a private cloud or have the capacity to implement and operate one.
In this section, we will delve into the details of these two experiments. Our goal is to provide an in-depth explanation of each experiment's design, methodology, and outcomes.

4.1. Prerequisite

For the experiments, we defined the following prerequisites: A local version of a LLM must be deployed on standard office hardware like standard developer laptops, utilizing open-source libraries such as PrivateGPT [3] or LocalGPT [4]. Due to security concerns, all sensitive data must remain within our internal network. This forbids the usage of cloud-based solutions and necessitates the implementation of a local LLM setup.
By fulfilling these prerequisites, we can ensure that the experiment is conducted in a secure and compliant manner, while also leveraging the capabilities of open-source libraries to support our operational needs.

4.2. Method

To conduct this experiment, we employed a combination of open-source libraries and "core technologies". We tested two open-source libraries, LocalGPT and PrivateGPT [3], to assess their suitability for our operational needs.
The core technologies used in this experiment include langchain, a Python library for building, testing, and deploying language models, as well as llamacpp, a C++ library for Large Language Model (LLM) development and deployment. We also utilized text embeddings from InstructorEmbeddings and SentenceTransformers to enhance model performance.
Furthermore, the models employed in this experiment were based on two distinct architectures: GPT4All-J, a variant of the GPT-4 model, and LlamaCpp (llama-2 based), a C++ implementation of the llama-2 model utilizing the llamacpp library.
By exploring these technologies, we aimed to create an experiment setup that could effectively evaluate the capabilities of open-source libraries in supporting our operational needs.

4.3. Architecture

The architecture of our experiments is depicted in Figure 1 (based on localGPT [4] and privateGPT [3] frameworks), which leverages state-of-the-art techniques for integrating Large Language Models (LLMs) with vector databases. The user initiates a query through the console, which triggers a similarity search in the vector database that stores our collection of documents, including flight procedures, manuals, and other relevant resources. The similarity search

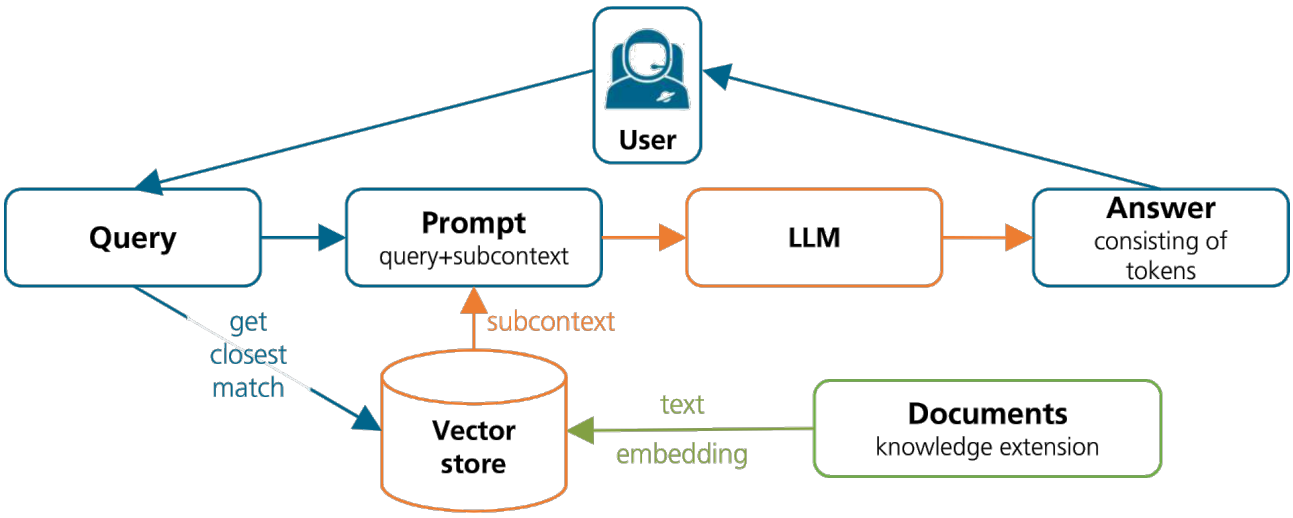

FIG 1. Experiment architecture: LLM with Vector Store.

is performed using a technique such as approximate nearest neighbor search (ANNS) or hierarchical navigable small world graphs (HNSW), which efficiently retrieves the most similar documents to the user's query. The resulting document set is then ranked based on their semantic similarity to the query, ensuring that the top-ranked documents are those that are most relevant and contextual. The output of this similarity search, along with the initial query, serves as the context for the LLM to generate an answer. This contextualized knowledge enables the LLM to not only draw upon its trained knowledge but also leverage the background information provided by our collection of documents. The LLM then uses this contextualized knowledge to build a response and returns it to the user, providing a detailed and relevant answer to their query.

4.4. Experiment I: Inference (Aug. 2023)

The first of our two experiments took place in August 2023. At that time, the open-source frameworks for running LLMs on local computers were still experimental, and we had to spend some time figuring out how to install and run a LLM locally on our laptops.

To gain knowledge about available frameworks, we decided to include this process as part of the experiment. We also wanted to see which framework would work best for us, so we quickly scanned the internet for guidance. This search proved helpful in getting started with our experiments.

Since we had limited time, we planned to split the experiment into three parts, each with a different but overlapping goal. Three people were involved, and we wanted to make sure everyone contributed their expertise.

The first part focused on inserting large documents into the LLM's context and using it to find certain facts within those documents. The second part aimed to use our internal wiki page as a resource for knowledge and reduce hallucinations (we'll discuss this in more detail later) within the LLM's answers. Finally, we wanted to test how to overwrite the LLM's trained facts with new facts injected into the LLM's context from a documents stored on our system.

Summarize Documents

In the first part of the experiment, we attempted to make sense of a large corpus of domain specific documents by summarizing overarching information or extracting a specific bits of information.

The input to our vector store were twofold: We ingested papers related to the topic of anomaly detection in the domain of space operations as well as user manuals and procedures of a visualization tool deployed at GSOC. As the experimenters have expert knowledge of both of these topics, the accuracy and completeness of the provided answers could be evaluated.

Additional hyper-parameters were tuned in this part of the experiment. One of the most important parameters here is the so called chunk size. In short, the chunk size parameter controls how many parts of ingested documents are utilized for answering a question. When, e. g., setting the chunk size to one, only a single document will be taken into account, which would make it impossible for the LLM to extract information from multiple documents. Setting the chunk size too high, however, worsens the performance and might lead to unrelated documents being considered for an answer. We achieved sufficient results with chunk sizes between 4 and 8 while still keeping the computational cost manageable.

For the evaluation of summarizing many documents at once, we asked the LLM to summarized the different approaches used for anomaly detection for satellite telemetry. Various different prompts were used and repeated in order to get a better understanding of the results. Overall, the results were very mixed. In many cases, the answer only mentioned a small amount of the provided approaches, sometimes even describing a single approach. While often being able to capture the general concept behind the anomaly detection methods, some of the descriptions also errors only detectable if one is already familiar with the topic.

When asking the LLM for a specific bit of information related to the ingested user manual and procedures, the results were overall better. While it sometimes denied an answer due to a perceived lack of information or provided useless answers, it was often able to name the correct procedure containing the requested information and give a short overview of its content.

In summary, the results of this part of the experiment show that there is still much to improve on and investigate. Getting incomplete or incorrect summaries of documents is a big problem for our operational use cases. While answering very specific question yielded better results, the number of

bad answers still need to be reduced to satisfy the expectation of our users.

RB-MBT Facts

As our goal is to support the training and operational needs of satellite engineers, this part of the experiment focused on ensuring the accuracy of the model outputs. For operational use, it is essential that our setup provides reliable information with minimal hallucination. Hallucination in LLMs refers to the phenomenon where a model generates responses that are entirely made-up or not supported by the input data, often due to overconfidence or a lack of understanding of the context. This can result in answers that are factually incorrect, irrelevant, or even absurd. Moreover, we require consistent answers to repeated queries, as varying responses can be problematic in critical applications.
To achieve these goals, we employed a vector store populated with internal wiki pages, which contain diverse formats and styles similar to those encountered in real-world satellite operations, such as:

- software documentation (single language, full sentences, many industry-specific abbreviations),
- meeting notes (mixture of German and English, mixture of bullet points and full sentences, both names and uncommon abbreviations in different formats),
- tabular information.

We conducted experiments to reduce hallucination and improve factual accuracy by varying hyperparameters and exploring different presets. In particular, we modified the following settings:

- `temperature`: adjusts the probability distribution of next tokens,
- `repetition_penalty`: penalizes tokens based on their frequency in the text and prompt,
- `top_k` and `top_p`: control the number and proportion of highest-probability tokens to retain.

We then asked the LLM questions about the content of those wiki pages and evaluated the accuracy of the answers. In addition, we compared the model outputs for repeated queries. While adjusting the listed hyperparameters improved output accuracy, we unfortunately still observed a significant share of both hallucinations and variability in responses to repeated prompts. These issues must be addressed in further experiments to ensure reliable performance in satellite operations.

Overwrite Trained Knowledge

In this part of the experiment, we examined how to persists and overwrites trained facts.
We designed this experiment to ensure the LLM trusts facts from the document store over those learned during its training. This is crucial because we can't control the learning and training phase, which involves large tech companies like Meta. During that phase, there's a risk that the LLM may learn false or inappropriate facts about space operations, since its training material often originates from the Web with only minor quality control.
In an operational use case, our operators need to rely on accurate information from the document store, not trained facts. We must ensure that our LLM is trustworthy and provides correct support based on those documents, not on knowledge it learned during its training at a AI company.
To test this, we used a technique called "prompt hacking" to prioritize facts from the document store over learned facts. We loaded a document into the document store containing several false facts, such as "Paris is the capital of Germany" or "The moon is made of cheese". These false fact should be way different to common facts, that we were sure the LLM had learned it differently during its training phase, e.g. that "Paris is the capital of France".
We then asked the LLM questions about these false facts and expected it to return the false fact instead of the correct one. Before delivering the question, we provided the false facts to the LLM's prompt which builds the context for the answers.
However, the results were varied: sometimes the LLM returned the false fact, sometimes it answered correctly based on its learned context, and occasionally it withdrew an answer, saying that the statement was not true.
For our operational use cases, this unpredictability is a problem. We need to be sure that facts from the document store (e.g., satellite documentation) are cited correctly and not overwritten by learned context.

Conclusion

In this experiment, we aimed to investigate the potential of LLMs in space operations. Despite being in the early stages of experimentation, we were able to achieve promising results within a short amount of time. Our first findings suggest that with further adjustments and refinements, an LLM could be developed into a viable product that meets our specific needs.
By exploring the capabilities of LLMs, we may be able to provide accurate information to operators and help them to operate more satellite more safely and economically. While there is still much work to be done, our experiment has demonstrated that LLMs can be a valuable tool to our engineers.

4.5. Experiment II: Application (Jan. 2024)

The second experiment took place in January 2024, six months after our initial study. The goal of the experiment was to deepen our knowledge about LLMs and to provide the Columbus FCT with an experimental tool to retrieve information fast during operations. This time, since all participants of the experiment used their gained knowledge of the last experiment, everyone setup a similar environment on its own laptop, injected data provided by the Columbus FCT into the document store and asked the LLM some predefined question.
This follow-up investigation leveraged significant technological advancements since then, including improved LLMs and more efficient frameworks for local installation on laptops.
The new models demonstrated substantially enhanced performance, generating answers at a much faster pace and with greater accuracy. Specifically, the incorporation of quantized models yielded better results compared to our earlier experiment.
The framework for local installation also underwent notable improvements, which was easier to install now and able to incorporate useful tools, such as automatic text extraction from images or tables, by utilizing Optical Character Recognition (OCR). This feature allows to seamlessly integrate technical documentation into the document storage that of-

ten include figures, schematics and tables with important texts.
Moreover, the improved frameworks can use the processing power of integrated Graphics Processing Units (GPUs), which significantly accelerates the answer-finding process and enables more natural conversations with the models.

Co-Pilot

After all participants had set-up their environments, we populated the LLM's document storage system with relevant engineering and operational documents provided by the Columbus FCT. Compared to satellite operations, the Columbus module of the International Space Station (ISS) is operated continuously 24/7, in order to provide real-time support to the astronauts at all times. Furthermore, ground controllers need to be alerted about and react to off-nominal events, such as emergencies. Those off-nominal events usually require a specific response from the ground controller in a extremely short manner. It is therefore vital for the Columbus FCT to acquire this knowledge during training. However, due to the complexity of the on-board systems and operations, flight controllers might not be able to memorize all pieces of information, but must focus their effort on the most critical pieces. This becomes even more evident, when looking at the document base, which was used for this experiment: From a total of nearly 400.000 documents available to the Columbus FCT, the used documents consisted of just 92 documents, all in .pdf format, covering user or operational manuals only. The Columbus FCT usually uses those documents to increase their Subject Matter Expertise (SME) in certain areas vital to operations during their certification, are used to write procedures for executing activities on-board Columbus or on ground, or are the basis for investigations into anomalous behavior of on-board equipment. The size of these documents ranged between a few pages, with the smallest documents being 14 pages long (91kB), and several thousand pages, with the largest documents being 8.700 pages long (180kB). This provided the LLM with the necessary context to understand the questions and topics being queried. Furthermore, this allowed us to gain insights into the performance during data ingestion and answer generation. All participants then posed a set of standardized questions, provided by the Columbus FCT, to their respective LLMs and recorded the responses. Finally, a member of the Columbus FCT independently reviewed and rated the answers generated by the LLMs against those provided by a human expert. The questions revolved around the on-board software of the Columbus Data Management Subsystem (DMS). A total of four questions were used.
The ratings were based on the following criteria: (i) “correct statements” if the content of the LLM's answer matched that of the human-provided response; (ii) “wrong statements” if the LLM's answer contained errors in terms of content; and (iii) “out of scope” if the LLM's response addressed a different topic or was unrelated to the question posed. The result of this evaluation is shown in Table 1, where we divided the number of evaluated statements by each participant (indicated by P1 through P4).
As can be seen in the table, the accuracy of the answers were below 50% in all cases. This is most likely due to the fact, that we used "out-of-the-box" models for this experiment, with minimal (or no) further tuning of hyper-parameters, no forms of prompt engineering and no re-training of the model.

Conclusion

In conclusion, with our experiment have made a first step to investigate the potential of LLMs in supporting space operations personnel with quick and reliable answers to complex technical questions. However, despite the promising results, it is essential to acknowledge that the performance of LLMs remains uncertain without further context and tuning.
The accuracy of LLM-generated answers varies significantly depending on the specific question posed, highlighting the need for parameter fine-tuning, prompt engineering, or even model re-training to optimize model performance for particular use cases.
Finally, it is crucial to recognize that comparability and repeatability are challenging to achieve and evaluate in LLM-based studies like ours. The inherent variability in model performance, question wording, and human evaluation make it difficult to draw definitive conclusions or generalize results across different contexts. Nevertheless, our research provides a solid foundation for future investigations into the capabilities and limitations of LLMs in space operations, and we hope that this work will contribute to a deeper understanding of their potential benefits and drawbacks.

5. FUTURE WORK

As a next step, in an on-going cooperation with the European Space Agency (ESA) in the course of their A2I Roadmap, we develop a LLM based tool for incident classification and root-cause analysis assistance. The A2I Roadmap aims to leverage ML to support incident classification and root-cause analysis in space mission operations, among other use cases. The roadmap identifies five priority domains and 14 specific use cases for targeted AI application development, including:

- Incident classification: Using ML to classify incidents based on their severity, impact, and cause
- Root-cause analysis: Utilizing ML to identify the underlying causes of incidents and anomalies in space mission operations

By applying ML techniques to these areas, the A2I Roadmap aims to enhance the efficiency and effectiveness of incident response and root-cause analysis in space mission operations, ultimately contributing to the growth and success of the European Space Sector.
With the Mars Exploration Telemetry-driven Information System (METIS), GOSC is currently developing an intelligent assistant for astronautical exploration missions into deep space. The assistant covers a broad area of operations, from monitoring to anomaly resolution, timeline planning, and activity execution. To achieve these functions, the assistant is divided into different agents, where each agent performs a specific function by using a specific machine learning or automation approach, as reported in [5]. For example, for the purposes of anomaly resolution, the so-called "Reasoning-Agent" relies on a Knowledge-Graph (KG), which combines different types of data from different sources. This KG might also be utilized by an LLM in order to generate text or reason about the knowledge that is contained within the graph. There

TAB 1. Results from the evaluation of queries.

	Number of correct statements P1/P2/P3/P4	**Number of false statements** P1/P2/P3/P4	**Number of out-of-scope statements** P1/P2/P3/P4	**Accuracy [%]**
Question 1	2 / 1 / 5 / 6 = 14	5 / 4 / 7 / 9 = 25	4 / 4 / 5 / 1 = 14	26%
Question 2	3 / 0 / 4 / 5 = 12	5 / 0 / 2 / 1 = 8	2 / 6 / 1 / 0 = 9	41%
Question 3	4 / 4 / 4 / 5 = 21	4 / 3 / 3 / 3 = 16	2 / 0 / 4 / 0 = 6	49%
Question 4	5 / 4 / 3 / n.a. = 12	3 / 4 / 4 / n.a. = 11	2 / 1 / 4 / n.a. = 7	40%

is currently an on-going project involving GSOC and the DLR institute for Software Technology, which will explore this possibility. First results will most likely be available by mid/end of 2025.

Building upon the success of our initial experiments, we plan to conduct a more comprehensive investigation into the rapid advancements in LLMs. Specifically, we aim to delve deeper into the realm of Retrieval Augmented Generation (RAG) techniques. The basic idea is to retrieve, i.e. fetch relevant information from a pre-trained index or database. And generate, i.e. use the retrieved information as input to generate new, coherent text.

Furthermore, we plan to integrate Ollama [6], a lightweight and extensible framework for building and running language models on local machines, into our experimental pipeline. By doing so, we aim to further streamline the development process and reduce computational overhead.

Lastly, we recognize the importance of optimizing vector database performance in the context of LLMs. We intend to conduct an in-depth exploration of this area, with a focus on developing novel indexing strategies and query optimization techniques to enhance overall model efficiency.

6. CONCLUSION

These experiments have made a first step towards investigating the potential of Large Language Models (LLMs) in supporting space operations personnel at GSOC with quick and reliable answers to complex technical questions. The results of our experiment are promising, but also highlight the need for further research to refine model performance.

The accuracy of LLM-generated answers varies significantly depending on the specific question posed, underscoring the importance of parameter fine-tuning, prompt engineering, or even model re-training to optimize model performance for particular use cases. We acknowledge that comparability and repeatability are challenging to achieve in LLM-based studies like ours, due to inherent variability in model performance, question wording, and human evaluation.

Despite these challenges, our research provides a solid foundation for future investigations into the capabilities and limitations of LLMs in space operations. By exploring the potential benefits and drawbacks of LLMs, we aim to contribute to a deeper understanding of their role in supporting operators and engineers in space-related tasks. Ultimately, this study demonstrates that LLMs have the potential to provide accurate information to operators, enabling them to operate satellites more safely and economically.

ACKNOWLEDGEMENTS

The authors are thankful to Steffen Zimmermann, and all members of the DLR MBT- and MIB-Teams for their valuable support and fruitful discussions.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN: 9781510860964.

[2] Hugging face. http://https://huggingface.co/. Accessed: 2024-09-06.

[3] Zylon by PrivateGPT. Privategpt, may 2023. https://github.com/zylon-ai/private-gpt.

[4] PromtEngineer. localgpt, may 2023. https://github.com/PromtEngineer/localGPT.

[5] Carsten Hartmann, Franca Speth, Dieter Sabath, and Florian Sellmaier. Metis: An ai assistant enabling autonomous spacecraft operations for human exploration missions. In *Proceedings of the 2024 IEEE Aerospace Conference*, Big Sky, MT, USA, 2024. IEEE. ISBN: 9798350304626.

[6] Ollama. Ollama, july 2023. https://github.com/ollama/ollama.