Table of Content

Overview **2**

Future work discussion **3**

Step-by-Step Workflow **4**

Human Prompt vs. System Prompt **6**

Retrieving Docs Filtering **7**

Framing and Chain-of-Thought Reasoning: Enhancing Document Utilization in RAG **10**

Raw Results to Confidence Level **11**

Results **12**

Mistral 7B 12

Orca 2 7B 19

Llama 2 13B 25

Gemma 2 2B 31

Custom corpus Testing Results 38

Custom GPT-4 corpus **41**

No relevancy Mistral 7B results **45**

Overview

In this work, we present our Retrieval-Augmented Generation (RAG) framework, designed to improve model performance on multiple-choice question (MCQ) System Engineering benchmark. Our system demonstrates improved accuracy across various tested models, including **Mistral 7B**, **Orca 2 7B**, **Llama 2 13B**, and **Gemma 2 2B**, on benchmark questions sampled at regular intervals. The corpus utilized for retrieval consists of the following minute assortment of domain-specific documents:

- Wikipedia entree: Systems engineering.
- USA Department of Defence: Systems Engineering Guidebook.
- MITRE: Systems Engineering Guide.
- DAU: Systems Engineering Fundamentals.
- INCOSE: SE Handbook, Version 2a.
- Systems Engineering Body of Knowledge (SEBoK).

Compared to typical industry datasets, which are significantly larger, our corpus is relatively small and narrowly focused. Achieving results with this dataset demonstrates the retrieval system's capability to work effectively with concise, targeted information. However, the limited size of the corpus does present challenges, such as reduced diversity and coverage, which can lead to gaps in information. Future work could address these limitations by incorporating additional domain-specific documents, crowd-sourced annotations, and other sources to expand and enrich the dataset.

A key challenge identified in our system was the retrieval of excessive irrelevant data, which hindered the model's ability to generate accurate predictions. To address this, we developed a **custom corpus** using GPT-4, containing concise, question-relevant information. This approach significantly improved performance, particularly for the Llama,Orca and Gemma models. Additionally, we integrated **"LLM as a Judge"** technique, where a secondary LLM evaluates and filters retrieved documents to ensure their relevance to the query. Furthermore, we incorporated **Framing** and **Chain-of-Thought (CoT) Reasoning** as part of our RAG framework.

This report discusses the following topics in detail:

- **Step-by-Step Workflow**: A detailed breakdown of our RAG system's architecture and execution.
- **Retrieving Docs Filtering**: Methods to evaluate and filter retrieved documents for relevance and quality.
- **RAG specific Framing and Chain-of-Thought Reasoning**: Techniques to enhance document utilization within the RAG framework.
- **Raw Results to Confidence Level**: How results are analyzed and confidence levels are established for predictions.
- **Benchmark Results**: Comparative performance of the models with and without RAG.
- **Our Custom Corpus**: Creation, testing, and the impact of a concise and question-specific dataset.

Our results demonstrate that incorporating RAG can enhance accuracy in answering graduate-level Systems Engineering questions.

Future work discussion

To further enhance our retrieval system, we are suggest focusing on several key areas:

- **Reducing Noise**: By improving data corpus chunking, we aim to meticulously label data chunks and develop a robust scoring system to ensure their relevance.
- **Improving Relevance Judgment**: This involves summarizing data chunks and refining our prompt engineering techniques to ensure that only the most relevant information is retrieved.
- **Exploring Generalized Systems**: We aim to design a retrieval system that is not tailored to specific models but instead works effectively across a wide range of models, ensuring flexibility and adaptability.
- **Expanding the Dataset**: Our current dataset is relatively small, and we plan to significantly extend it by incorporating a diverse range of data relevant to systems engineering. This expansion is intended to help our system handle a wider variety of queries and address more complex scenarios, improving its accuracy and utility.

  While not the main purpose of this work, there is potential to develop an open-source System Engineering vector database (vectorDB) in the future. Such a resource could be valuable for those looking to specialize their models in systems engineering. This vectorDB might include a variety of materials, such as articles, academic papers, books, system designs, and technical drawings, forming a comprehensive knowledge base. Although this concept is still in its early stages, it offers a compelling direction for enhancing the availability of high-quality resources and supporting broader advancements in the field.

- **Real-World Testing**: We would like to evaluate the system in real-world, day-to-day systems engineering tasks, moving beyond benchmark-specific testing. This will help us assess its practical utility and performance in real-life scenarios.

Step-by-Step Workflow

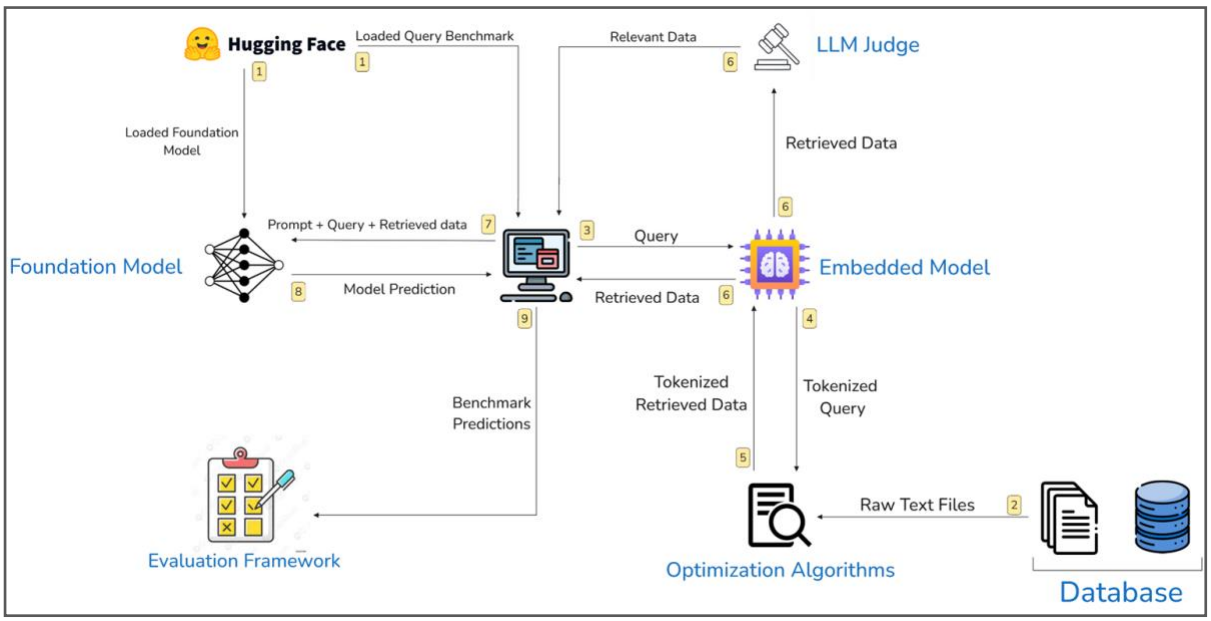


Fig. 1: Workflow Diagram. This diagram outlines the RAG system's process, starting with loading the foundation model and leading through data retrieval, prediction, and evaluation stages.

1. Loading the Foundation Model

- The system begins by loading a **foundation model** from **Hugging Face**.
- A **benchmark dataset** is also loaded, providing queries to test the performance of the models with and without the use of the RAG framework.

2. Database Integration

- A **domain-specific database** of raw text files is accessed. These documents form the corpus for retrieval.
- The raw database is **preprocessed** using:
  - **Chunking algorithms** to divide large documents into manageable segments.
  - **Relevance filtering** to remove redundant or low-value chunks.

3. Query Input

- **Queries** are selected from the **benchmark dataset** and passed into the system for processing.

4. Vectorization

- Both the **queries** and the **preprocessed database** are vectorized using a semantic embedding model with the **database** saved as a **ChromaDB's vector store**.

5. Document Retrieval

- The system performs a similarity search between the **query embedding** and the **database embeddings** to identify the **top-k most similar documents**.
- Retrieved documents are augmented with **Framing** and **Chain-of-Thought Reasoning** instructions to enhance their relevance and guide the model's structured reasoning.

6. Relevance Judging (Optional)

- The top-k retrieved documents are evaluated for relevance:
    - **Without Judge**: The retrieved documents are returned directly to the main program for use in the next step.
    - **With Judge**: The top-k documents are passed to an **LLM Judge**, which determines their relevance to the query. Only the judged relevant documents are returned to the main program.

7. Prediction

- The **query**, **retrieved data**, and **prompt instructions** are provided to the foundation model for generating predictions.

8. Output

- The foundation model returns its **prediction** based on the query and retrieved documents.

9. Benchmark Evaluation

- The system evaluates the model’s predictions against the benchmark dataset to measure accuracy and overall performance.

Human Prompt vs. System Prompt

A Human Prompt is the input directly provided by a user to the system, such as a question, query, or instruction describing the task to be performed. It is static, manually crafted, and serves as the starting point for the system's processing. In contrast, a System Prompt is an automatically generated input constructed by the system "behind the scenes." It combines the human prompt with additional context, such as retrieved documents or predefined instructions, to optimize the input for the model. While the human prompt defines the problem, the system prompt refines and structures it to enhance the model's ability to generate accurate and relevant responses.

How this works in our framework:
The **Human Prompt** in our framework is the query provided by a user or in our case the System Engineering benchmark. It defines the task or question for the system to address. For example:

- *"What are the key principles of Systems Engineering?..."*
- *"How does risk management contribute to Systems Engineering processes?..."*

These queries are simple and contain no additional context or instructions—they initiate the retrieval and reasoning process.

The **System Prompt** is dynamically generated by our framework based on the human prompt. It includes:

1. **The Human Prompt**: The core query or task.
2. **Retrieved Documents**: Relevant text retrieved from the corpus using semantic similarity and a LLM as a judge.
3. **Predefined Instructions**: Additions like **Framing** or **Chain-of-Thought (CoT) Reasoning** to guide the model's reasoning process.

Retrieving Docs Filtering

Various strategies are employed to improve the effectiveness of the Retrieval-Augmented Generation (RAG) system in managing noise. These strategies focus on controlling data retrieval and applying filtering techniques to ensure that only relevant information is processed.

```python
# Precompiled regex patterns
NAVIGATION_PATTERN = re.compile(r'<.*?>')
VERSION_METADATA_PATTERN = re.compile(r'SEBOK v\.\s*\d+\.\d+\.\d+|released \d{1,2} \w+ \d{4}', re.IGNORECASE)
FIGURE_PATTERN = re.compile(r'Figure\s+\d+.*|^Page\s+\d+$', re.IGNORECASE)
REFERENCES_PATTERN = re.compile(r'^References\s*\nNone\.$', re.IGNORECASE)
DOTS_DASHES_PATTERN = re.compile(r'[\.\-]{5,}')
ENDS_WITH_NUMBER_PATTERN = re.compile(r'[\s\.\-]\d+\s$')
NUMBERING_PATTERN = re.compile(r"^\d+(\.\d+)*\s")
BIBLIOGRAPHIC_PATTERN = re.compile(
    r'(ISBN|pp\.|Ver\.|Technical Report|Handbook|Edition|CRC Press|Taylor & Francis|MITRE|INCOSE)', re.IGNORECASE
)
STOP_WORDS = {"the", "of", "and", "to", "in", "on", "with", "a", "is"}

def is_potentially_irrelevant(text, title_keywords=None):
    if title_keywords is None:
        title_keywords = ["chapter", "contents", "table of contents", "section", "index"]

    # Check for navigation hints and version metadata
    if NAVIGATION_PATTERN.search(text) or VERSION_METADATA_PATTERN.search(text):
        return True

    # Check for figure labels, page numbers, and empty references
    if FIGURE_PATTERN.search(text) or REFERENCES_PATTERN.search(text):
        return True

    # Check for a high number of dots or dashes
    if DOTS_DASHES_PATTERN.search(text):
        return True

    if sum(c == '.' for c in text) / len(text) > 0.2 or sum(c == '·' for c in text) / len(text) > 0.2:
        return True

    # Check if the line ends with a number
    if ENDS_WITH_NUMBER_PATTERN.search(text):
        return True

    # Check for bibliographic patterns
    if BIBLIOGRAPHIC_PATTERN.search(text):
        return True

    # Check for common title or TOC keywords
    if any(keyword.lower() in text.lower() for keyword in title_keywords):
        return True

    # Exclude very short text with numbering
    if NUMBERING_PATTERN.match(text):
        return True

    # Check for excessive capitalization
    words = [word for word in text.split() if word.lower() not in STOP_WORDS]
    capitalized_words = sum(1 for word in words if word.isupper())
    if len(words) > 0 and capitalized_words >= len(words) / 2:
        return True

    # Exclude chunks that are too short or too long
    if len(text.split()) < 15 or len(text) > 1000:
        return True

    # If none match, consider it relevant
    return False
```

Fig. 3: Python Script for Text Relevance Assessment. This script utilizes various regular expressions and logical conditions to determine the irrelevance of text based on content and formatting criteria.

The system segments the dataset into chunks, which are then stored in a vector database for efficient retrieval. By adjusting the number of chunks retrieved, it is possible to balance knowledge diversity with noise reduction.

```python
# Retrieve top-k documents for the query in this case k =4
retriever = vector_store.as_retriever(
    search_type="similarity",
    search_kwargs={'k': 4}  # Number of documents to retrieve
)
retrieved_docs = retriever.invoke(query)
```

Fig. 3: Code Snippet for Document Retrieval. This Python code initializes a retriever to find the top-4 most similar documents to a given query, showcasing the use of similarity-based search parameters.

A large language model (LLM) is used to filter out irrelevant chunks from the retrieval set. This helps reduce noise and ensures that the data being processed is of high quality, which improves the accuracy and reliability of the outputs.

```python
def check_relevance_with_gradual_scores(document, query):
    prompt = f"""
    You are a systems engineering assistant. Assess if the retrieved document helps answer the query.
    Options: Relevant, Partially Relevant, Irrelevant.

    Query: "{query}"
    Document: "{document.page_content}"
    """
    response = client.chat.completions.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "system", "content": "You are an expert assistant evaluating relevance."},
            {"role": "user", "content": prompt}
        ],
        temperature=0
    )
    relevance = response.choices[0].message.content.strip().lower()
    return relevance in ["relevant", "partially relevant"]
```

Fig. 3: Relevance Assessment Function. This Python function checks the relevance of a document to a query using a system-engineering assistant role within an AI model, determining if the document is relevant or partially relevant based on the content interaction.


```
{"query": "Question: What methodologies are commonly used in the risk identification phase of systems engineering projects?\nA. Relying solely on past experiences without consulting current project data or team input.\nB. Independent assessments, SOW requirements analysis, brainstorming sessions with SMEs, interviews with IPT leads, review of similar/historical programs, trade studies, and review of technical performance measurements and project management data.\nC. Exclusively using automated tools to predict risks without human expert input or analysis of current project specifics.\nD. Limiting risk identification to financial risks while ignoring technical, schedule, and operational risks.\nAnswer:", "ctxs": [{"retrieval text": "Imagine you are a Systems Engineering expert. Based on the retrieved information, evaluate the following carefully. Analyze each piece of information step-by-step to answer the question:\nidentified risks within the systems engineering process. The risk management process requires the involvement of\nseveral disciplines and encompasses empirical, quantitative, and normative judgmental aspects of decision-making.\nFurthermore, risk assessment and management should be integrated and incorporated within the broader holistic\napproach so technology management can help align the risk management requirements to the overall systems\nengineering requirements. Thus, the inclusion of a well defined risk management plan that deals with the analysis of\nrisks, within the systems engineering master plan is vital for the long term and sustained success of any system\n(Blanchard and Fabrycky 2011).\n"}, {"retrieval text": "Based on the retrieved information, evaluate the following carefully. Analyze each piece of information step-by-step to answer the question:\nmanagement. Hillson(2003), Olsson (2007), and Chapman and Ward (2003) provide highly cited introductions.\nAdditional information on risk management for systems engineering projects can be found in the Risk Management\narticle in Part 3: Systems Engineering and Management.\n"}]}
```


Fig. 4: Example query from retrieved docs json file. Question: “What methodologies are commonly used in the risk identification phase of systems engineering projects?” relevant retrieved text: “Imagine you are a Systems Engineering expert. Based on the retrieved information, evaluate the following carefully. Analyze each piece of information step-by-step to answer the question: identified risks within the systems engineering process. The risk management process requires the involvement of several disciplines and encompasses empirical, quantitative, and normative judgmental aspects of decision-making. Furthermore, risk assessment and management should be integrated and incorporated within the broader holistic approach so technology management can help align the risk management requirements to the overall systems engineering requirements. Thus, the inclusion of a well-defined risk management plan that deals with the analysis of risks, within the systems engineering master plan, is vital for the long-term and sustained success of any system (Blanchard and Fabrycky 2011).” noise retrieved text: “Based on the retrieved information, evaluate the following carefully. Analyze each piece of information step-by-step to answer the question: Hillson (2003), Olsson (2007), and Chapman and Ward (2003) provide highly cited introductions. Additional information on risk management for systems engineering projects can be found in the Risk Management article in Part 3: Systems Engineering and Management.”

This example shown in fig.4, of retrieved text, highlights how different models can respond differently to the same system parameters. The first chunk provides actionable insights into risk management methodologies, while the second references external sources without detailing specific methods. While the selection process prioritizes relevance, some models may

handle additional length and minor noise better than others, emphasizing the need to tailor the system to the model's characteristics.

The evaluation system also allows for control over the number of chunks retrieved per query, enabling a balance between retrieving comprehensive information and noise reduction, based on the specific needs of the query.

```
# Mistral as example:
!lm_eval --model hf \
    --model_args pretrained=$Mistral_7B \
    --include_path $INCLUDE_TASK_PATH \
    --tasks $OUR_TASK \
    --retrieval_file "{RETRIEVED_DOCS}" \
    --concat_k 2 \ # decides how many retrieved docs are added to system prompt.
                   # this case it added the top 2 retrieved docs for this query
```

Fig. 5: LM Evaluation Command. This command line script showcases how to evaluate a language model using specific model arguments, tasks, and a retrieval component to include the top two documents relevant to a query in the system prompt

Although the system works within a specific domain and relies on a small dataset, noise remains a challenge. The effectiveness of noise management strategies can vary depending on the model's architecture and the task at hand, highlighting the need for tailored approaches to optimize performance.

Framing and Chain-of-Thought Reasoning: Enhancing Document Utilization in RAG

Our RAG framework Introduces two techniques, **Framing** and **Chain-of-Thought (CoT) Reasoning**, aimed at addressing challenges with retrieving and utilizing complex or irrelevant data. These techniques work in tandem with our retrieval system, ensuring that the retrieved documents are effectively leveraged by the models. By optimizing how retrieved information is structured and contextualized, the framework enhances the models' ability to process and apply this data to benchmark questions.

Framing: Providing Domain Context to the Model

Framing involves explicitly contextualizing queries or inputs to guide the language model's reasoning within a specific domain. This technique is particularly effective when paired with retrieved documents, as it "primes" the model to interpret and apply the retrieved information as a subject-matter expert. Framing reduces ambiguity in the retrieved data and focuses the model's attention on relevant domain knowledge.

In our framework, framing is introduced through a static instruction added to the retrieved data:

$$framing\_context = "Imagine\ you\ are\ a\ Systems\ Engineering\ expert."$$

Chain-of-Thought Reasoning: Structured Query Analysis

Chain-of-Thought (CoT) Reasoning is a technique where the model is guided to break down its reasoning process into intermediate steps before arriving at a final answer. When applied to retrieved documents, this technique ensures that the model systematically evaluates each piece of information, enabling more robust and accurate reasoning.

In our framework, CoT reasoning is introduced through a separate contextual instruction appended to each retrieved document aimed at emphasizing step-by-step analysis:

$$cot\_reasoning\_context = "Based\ on\ the\ retrieved\ information, evaluate\ the\ following\ carefully.\ Analyze\ each\ piece\ of$$

Both **Framing** and **Chain-of-Thought (CoT) Reasoning** are automatically incorporated into the **system prompt**. These instructions are dynamically added alongside the retrieved documents, ensuring the model receives clear guidance on how to interpret and process the information. By automating this process, the framework consistently applies domain-specific context and step-by-step reasoning to every query, enhancing the model's ability to generate accurate and relevant answers.

Raw Results to Confidence Level

In the lm_harness framework, the output for each question is provided as raw scores called logits, representing the initial assessment of each answer's likelihood. Logits are the raw outputs of the final layer in a neural network before any activation function like softmax is applied.

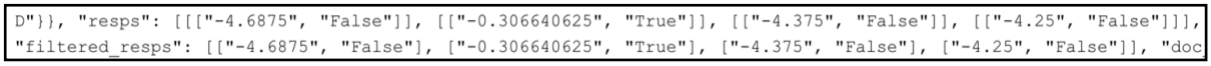


Fig. 6: LM Harness Evaluation Output. This image displays the logits and selection results from an LM harness evaluation. Choices [A, B, C, D] yielded logits of [-4.6875, -0.306640625, -4.375, -4.25], respectively, with choice B selected by the

These scores are directly processed using the softmax function, which converts them into a set of probabilities ensuring that the combined total is exactly one. The softmax function is mathematically represented as:

$$P(y = i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$$

where $P(y = i)$ is the probability of choice i, z_i is the logit corresponding to choice i, and N is the total number of choices. Subsequently, these probabilities are multiplied by 100 to convert them into percentage confidence scores.

This method enhances numerical stability and clarity. By transforming logits into probabilities, it prevents numerical underflow or overflow, thereby ensuring more reliable outputs.

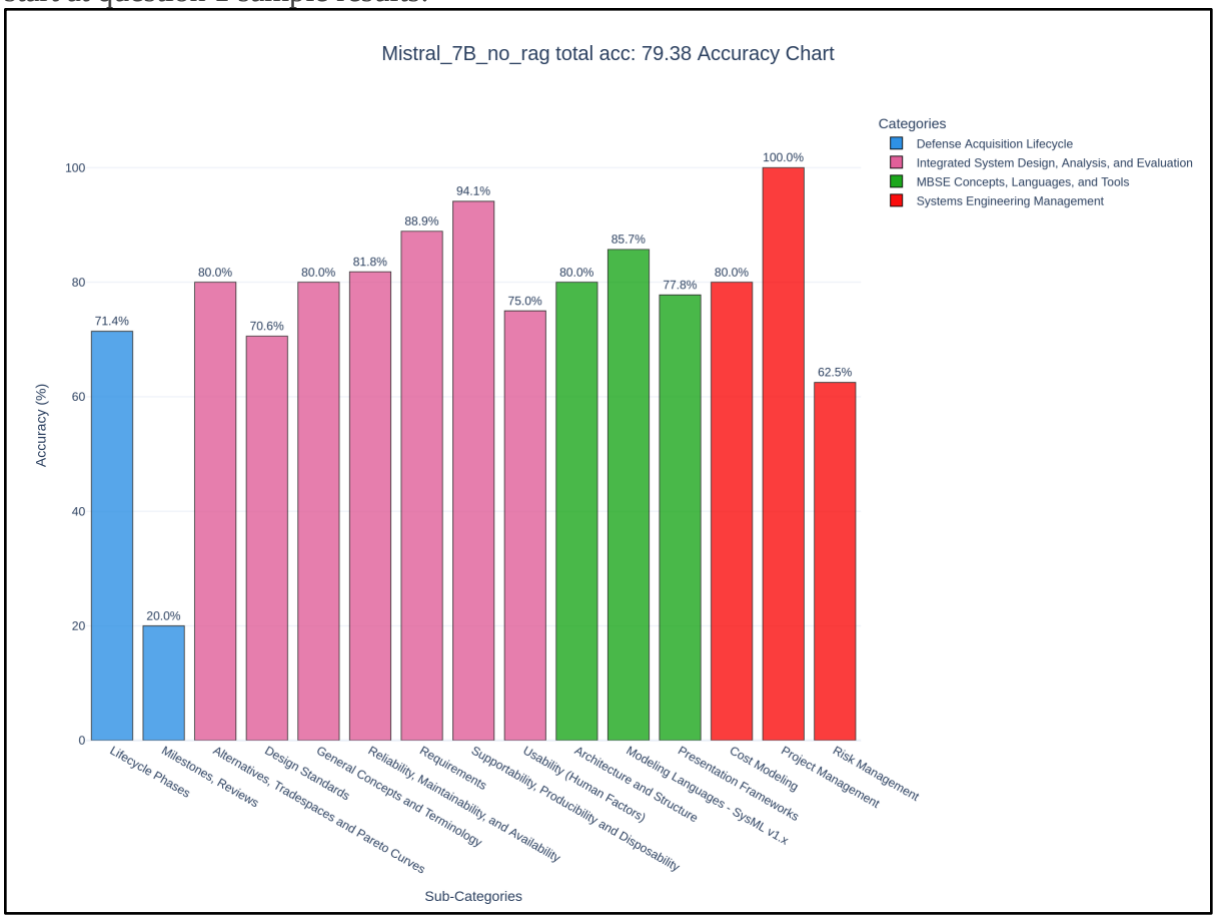
Results

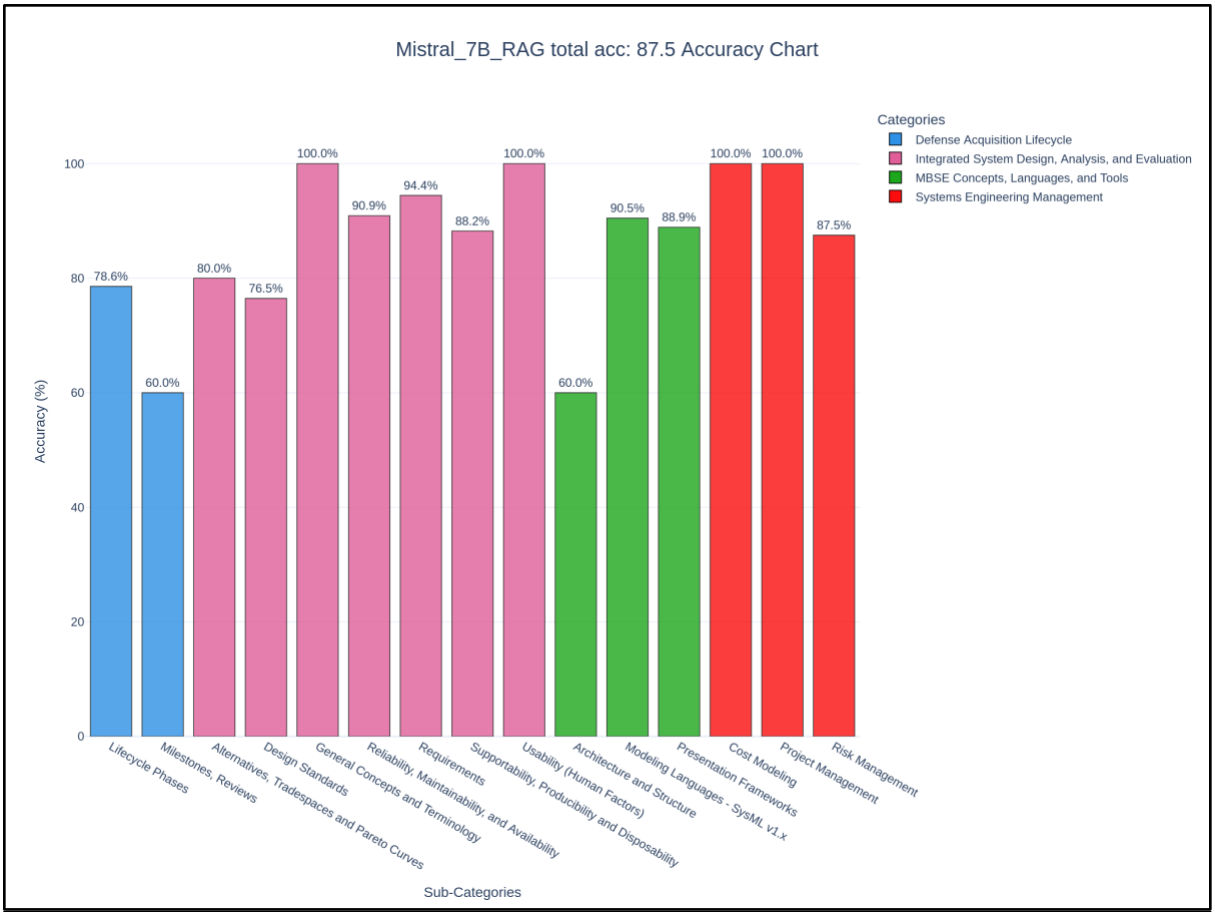
Each model's results were obtained using specific parameters, including the use or absence of an LLM judge and the number of retrieved documents, optimized to maximize accuracy.

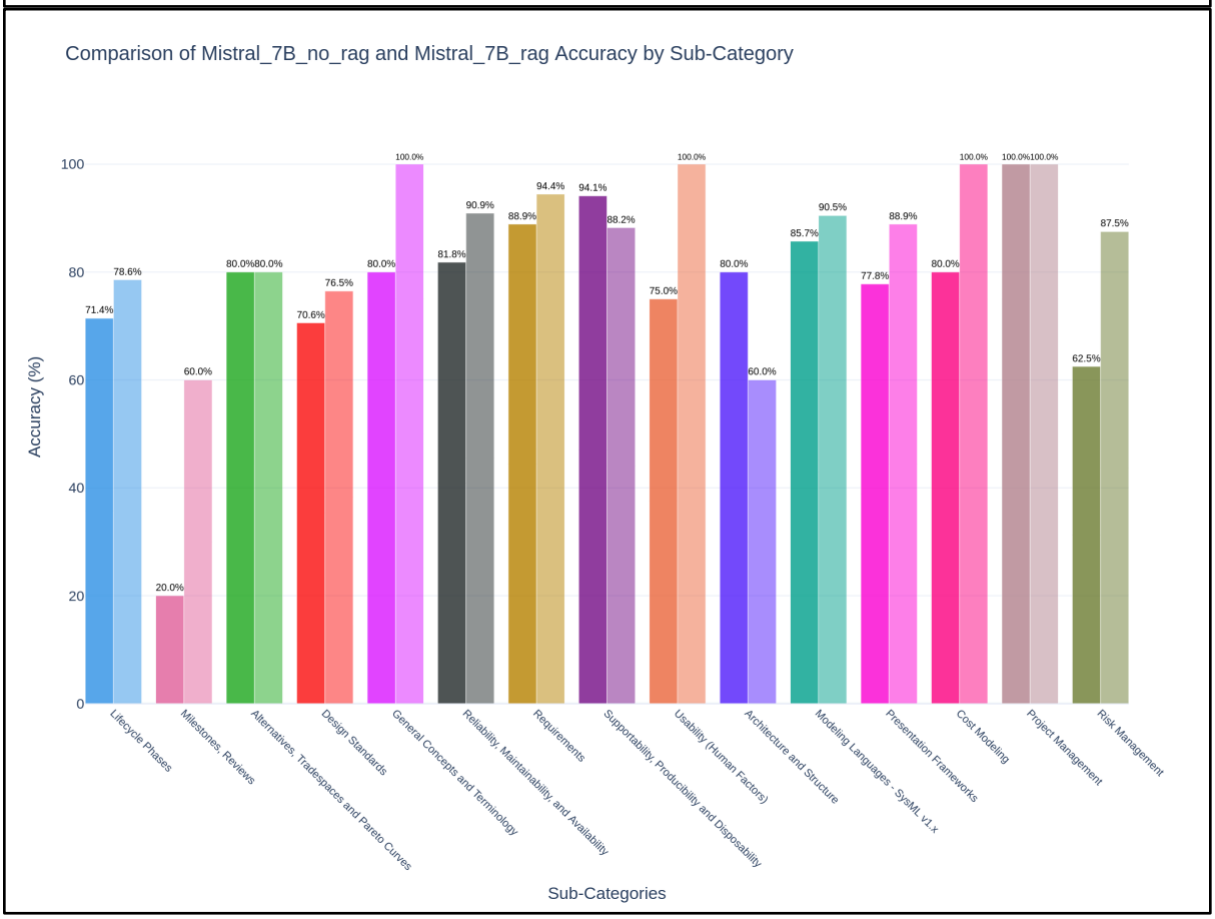
Mistral 7B

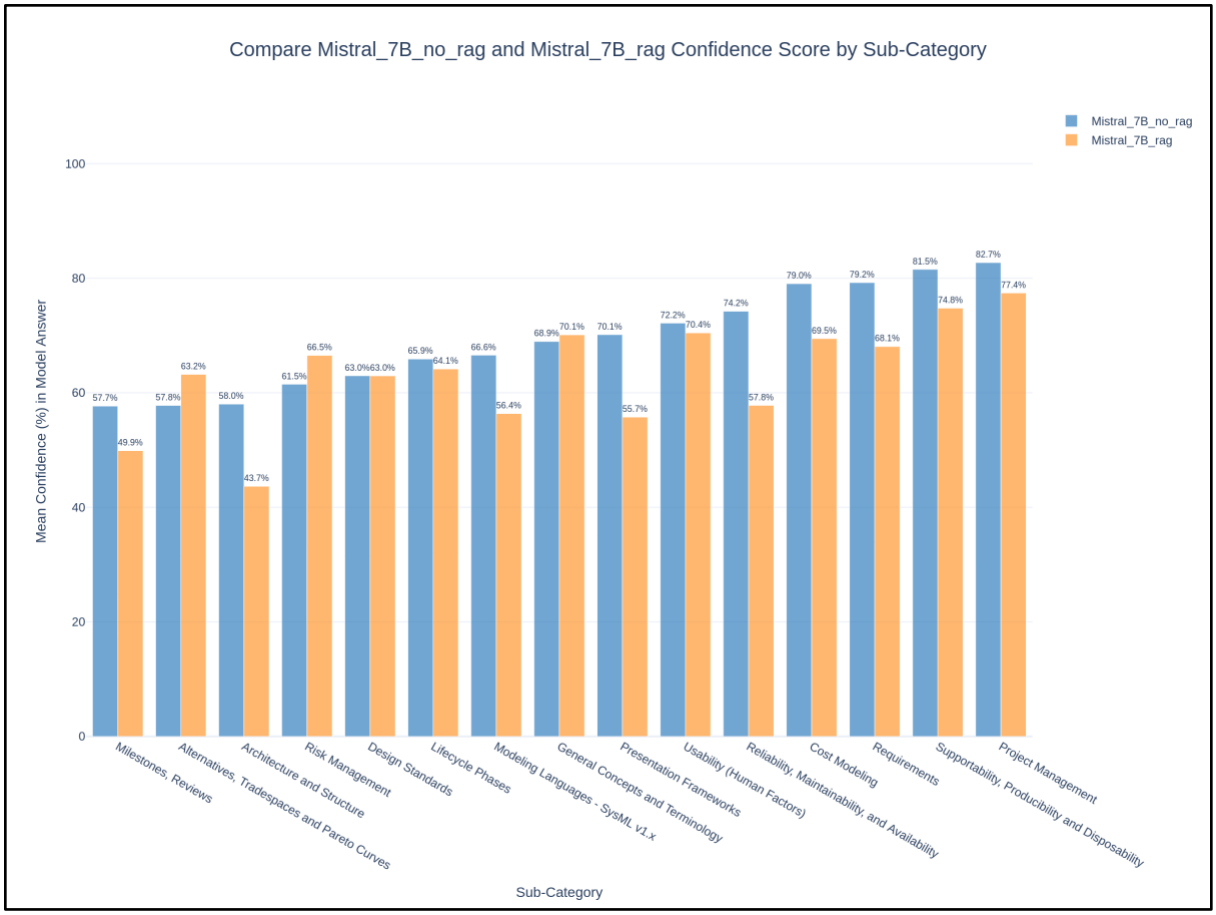
Mistral is our best performing model on the current version of our corpus and RAG Framework with an accuracy scores of 87.5% and 85% of each question set respectively. We were able to reduce the error rate by 39.38% on the first question set and by 14.29% on the second, we achieved the highest results using RAG with a LLM “as a judge” as well as retrieving the top 3 most relevant files, we believe the model is less sensitive to longer and denser prompts as long as they contain relevant text therefore benefiting from “a lot” of information this is also suggested by worse performs on the same corpus without relevance and from tests with less retrieved text.

start at question 1 sample results:

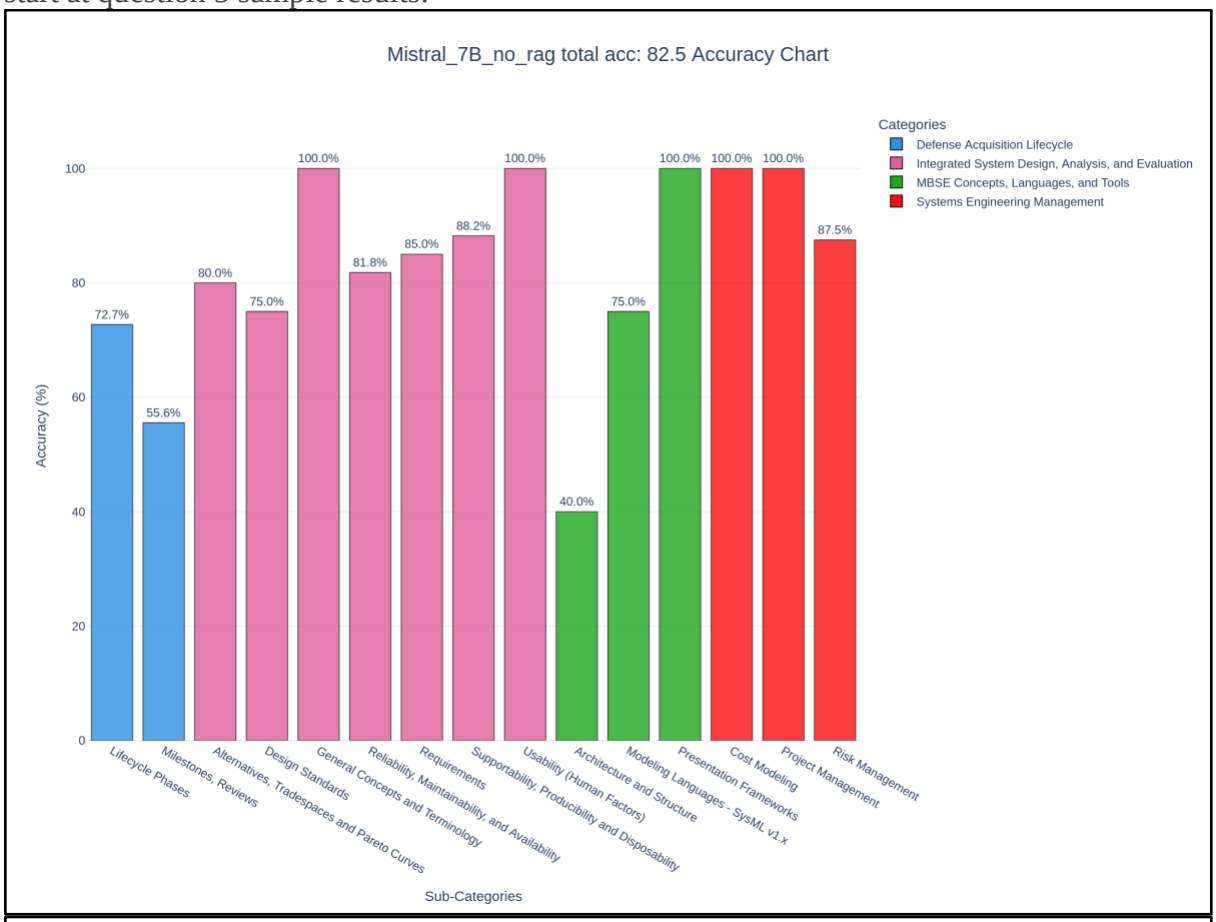


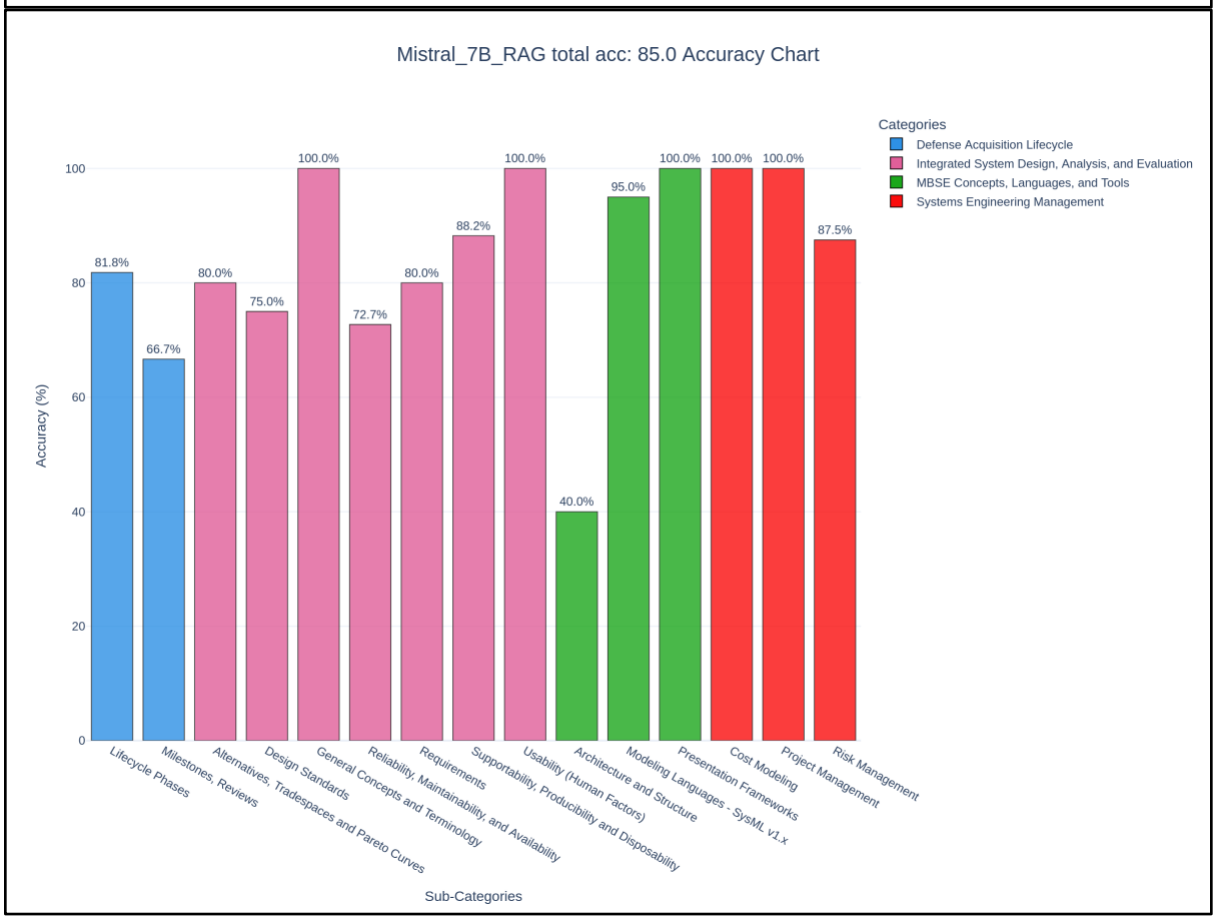

Mistral_7B_RAG total acc: 87.5 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
Sub-Categories
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management
78.6%
60.0%
80.0%
76.5%
100.0%
90.9%
94.4%
88.2%
100.0%
60.0%
90.5%
88.9%
100.0%
100.0%
87.5%

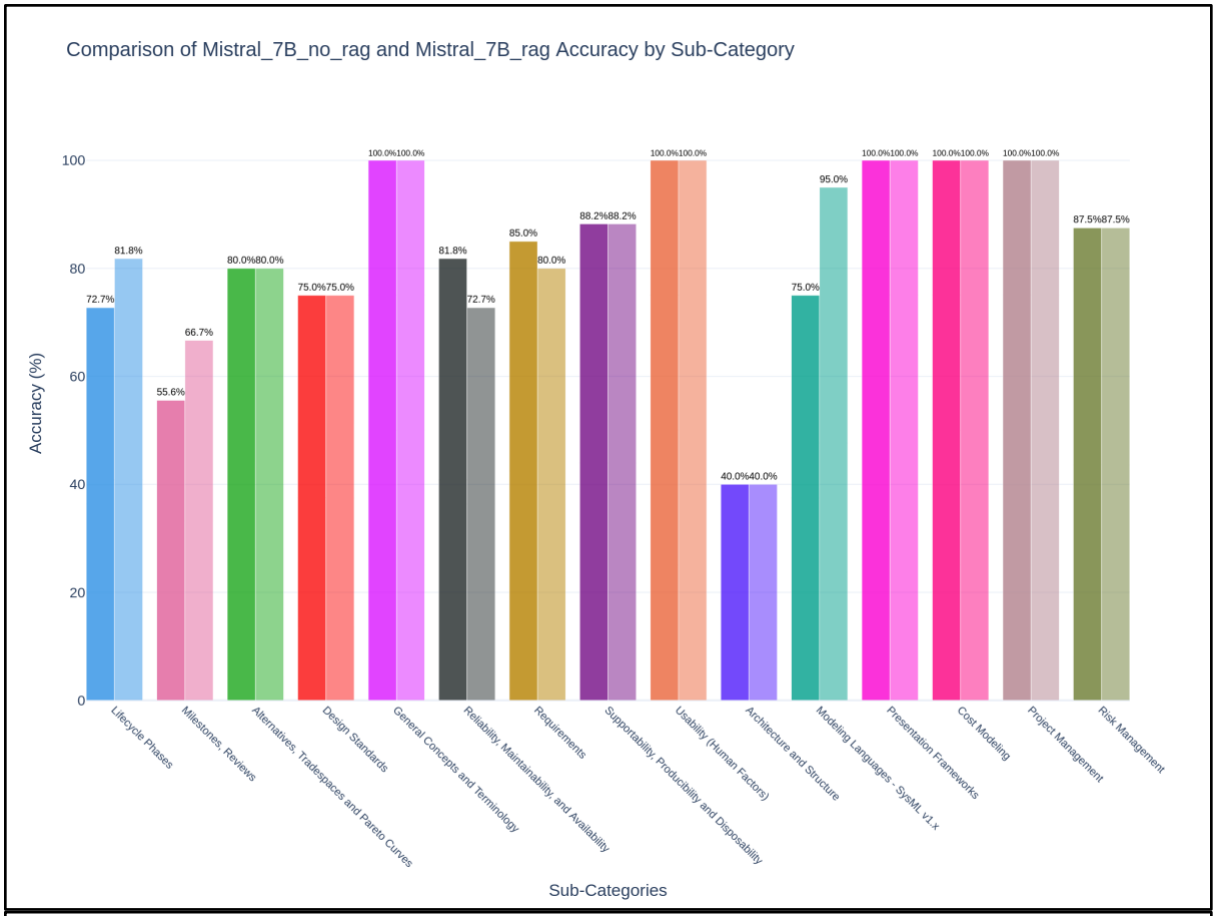

Comparison of Mistral_7B_no_rag and Mistral_7B_rag Accuracy by Sub-Category
Accuracy (%)
Sub-Categories
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management
71.4%
78.6%
20.0%
60.0%
80.0%
80.0%
70.6%
76.5%
80.0%
100.0%
81.8%
90.9%
88.9%
94.4%
94.1%
88.2%
75.0%
100.0%
80.0%
60.0%
85.7%
90.5%
77.8%
88.9%
80.0%
100.0%
100.0%
100.0%
62.5%
87.5%

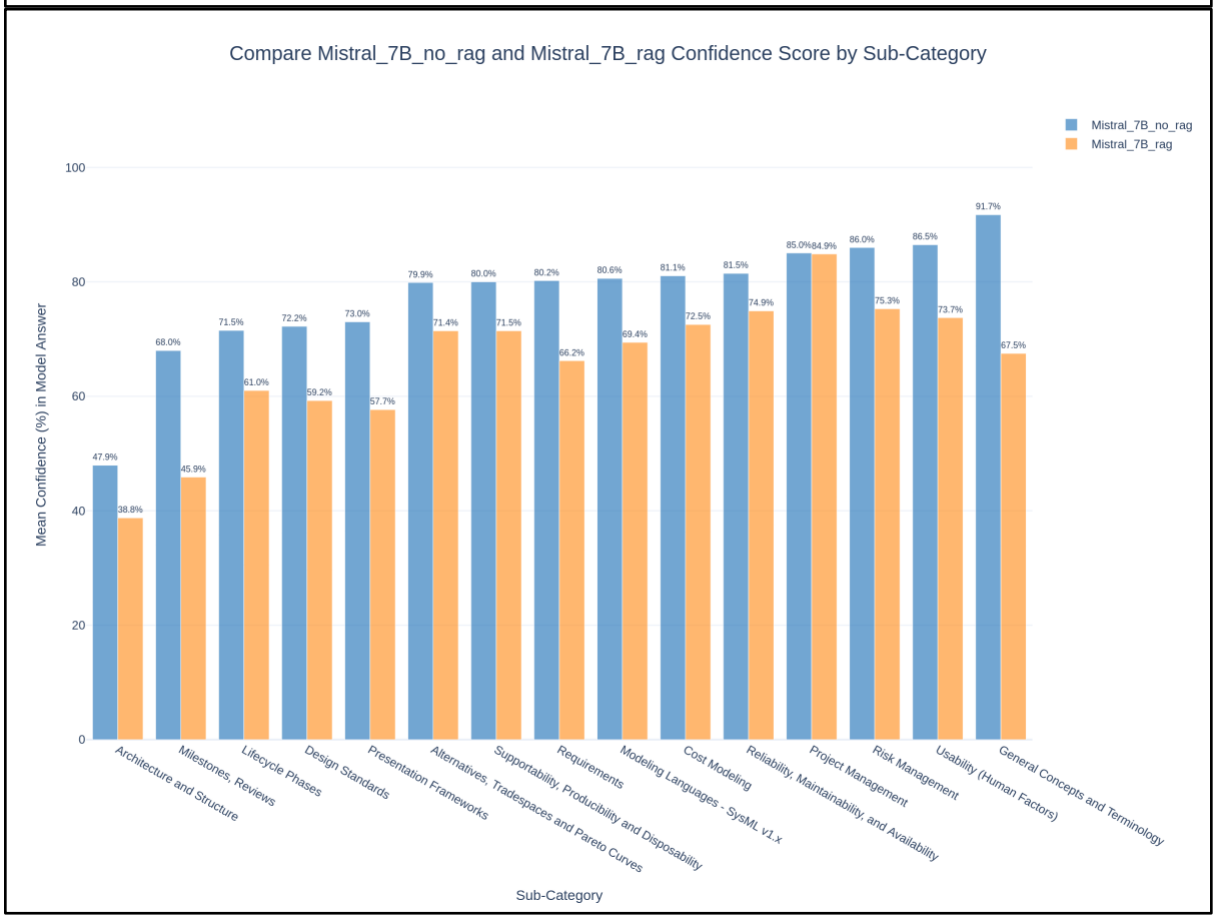

Compare Mistral_7B_no_rag and Mistral_7B_rag Confidence Score by Sub-Category
Mistral_7B_no_rag
Mistral_7B_rag
Mean Confidence (%) in Model Answer
100
80
60
40
20
0
57.7%
49.9%
57.8%
63.2%
58.0%
43.7%
61.5%
66.5%
63.0%
63.0%
65.9%
64.1%
66.6%
56.4%
68.9%
70.1%
70.1%
55.7%
72.2%
70.4%
74.2%
57.8%
79.0%
69.5%
79.2%
68.1%
81.5%
74.8%
82.7%
77.4%
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Architecture and Structure
Risk Management
Design Standards
Lifecycle Phases
Modeling Languages - SysML v1.x
General Concepts and Terminology
Presentation Frameworks
Usability (Human Factors)
Reliability, Maintainability, and Availability
Cost Modeling
Requirements
Supportability, Producibility and Disposability
Project Management
Sub-Category

start at question 5 sample results:





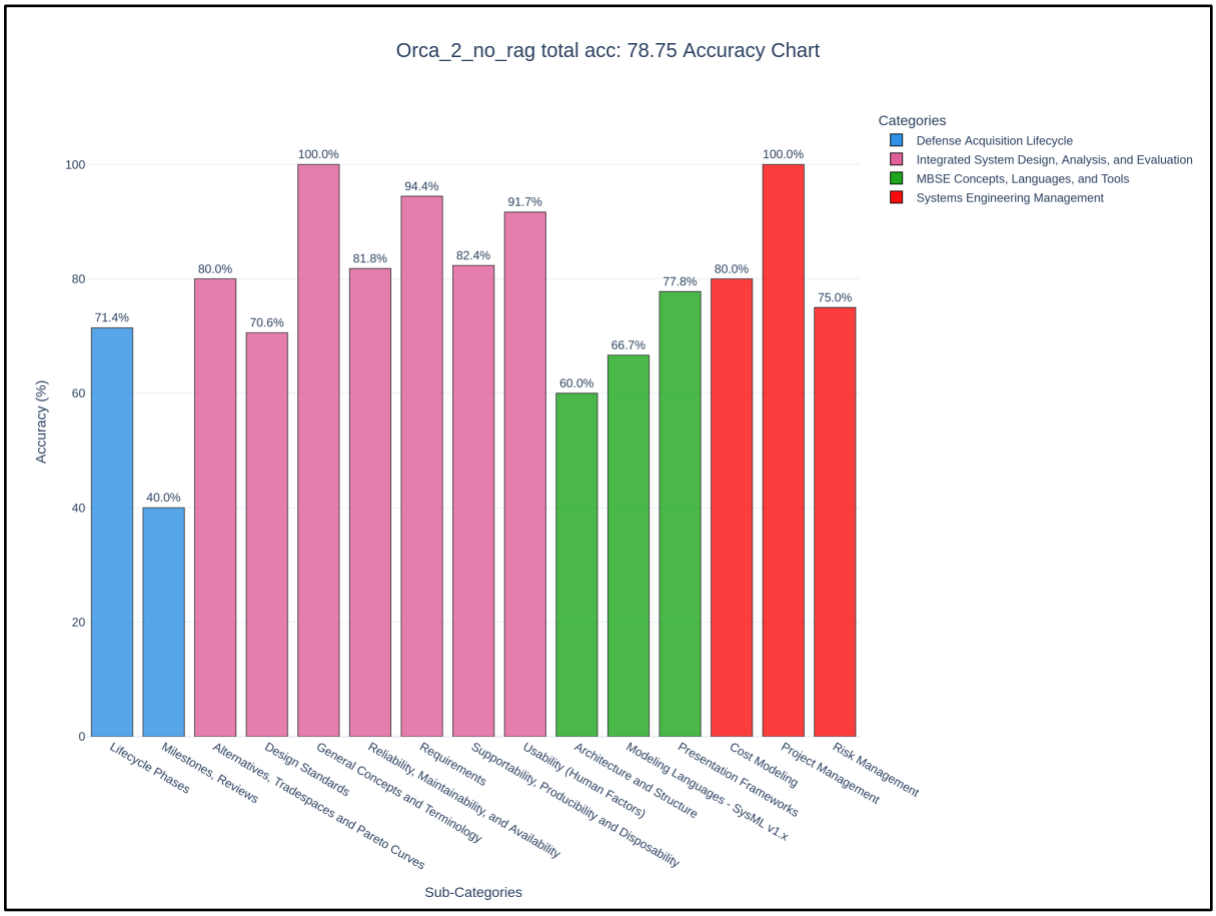

Comparison of Mistral_7B_no_rag and Mistral_7B_rag Accuracy by Sub-Category
Accuracy (%)
Sub-Categories
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management

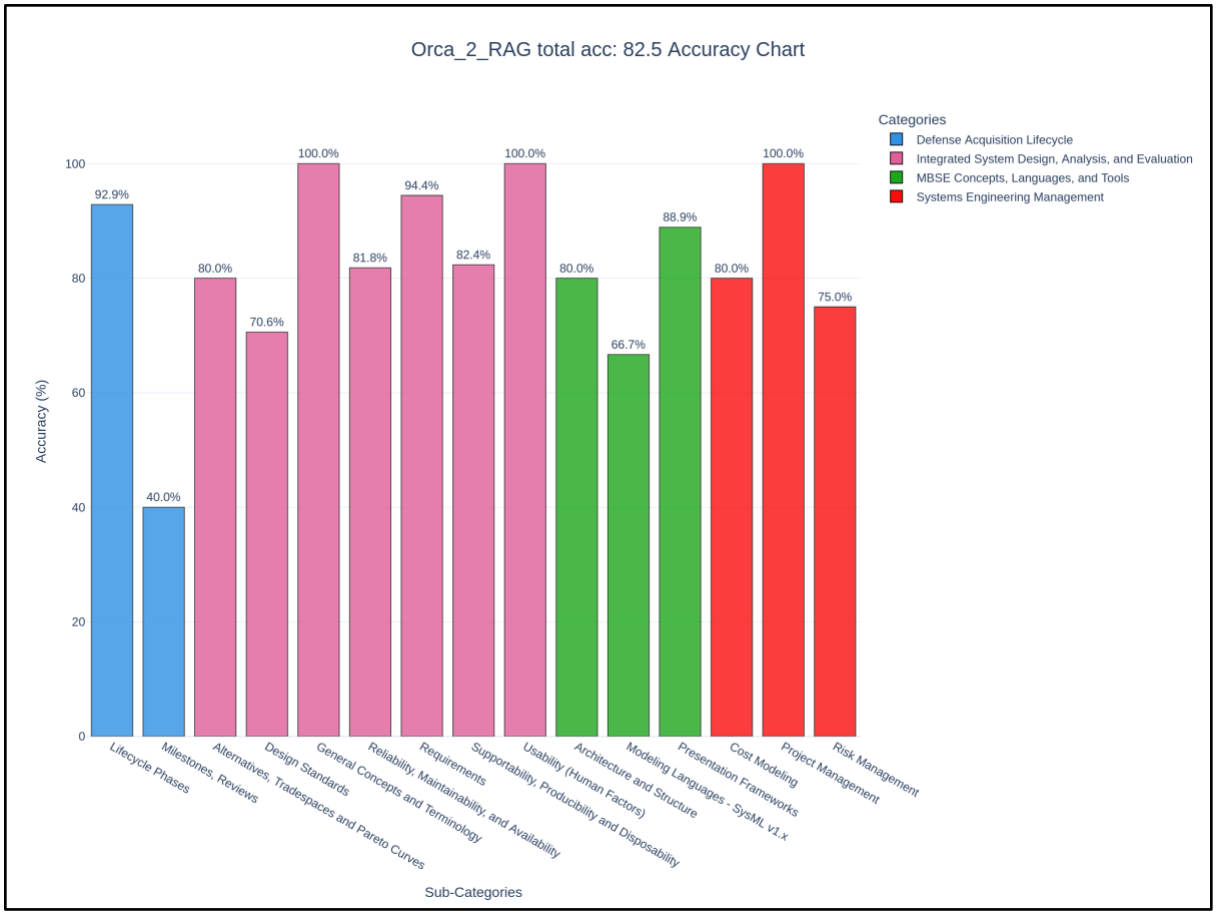

Compare Mistral_7B_no_rag and Mistral_7B_rag Confidence Score by Sub-Category
Mistral_7B_no_rag
Mistral_7B_rag
Mean Confidence (%) in Model Answer
Sub-Category
Architecture and Structure
Milestones, Reviews
Lifecycle Phases
Design Standards
Presentation Frameworks
Alternatives, Tradespaces and Pareto Curves
Supportability, Producibility and Disposability
Requirements
Modeling Languages - SysML v1.x
Cost Modeling
Reliability, Maintainability, and Availability
Project Management
Risk Management
Usability (Human Factors)
General Concepts and Terminology

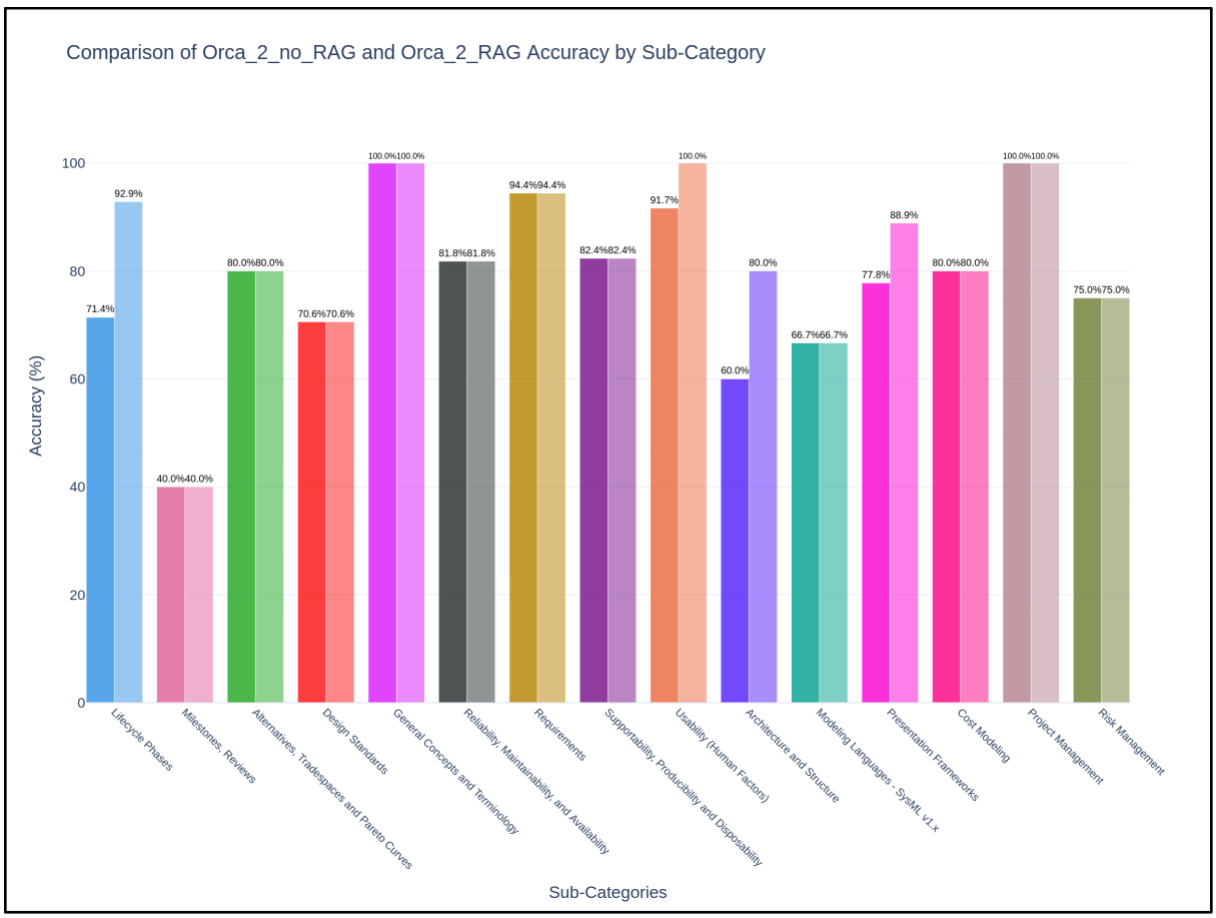
Orca 2 7B

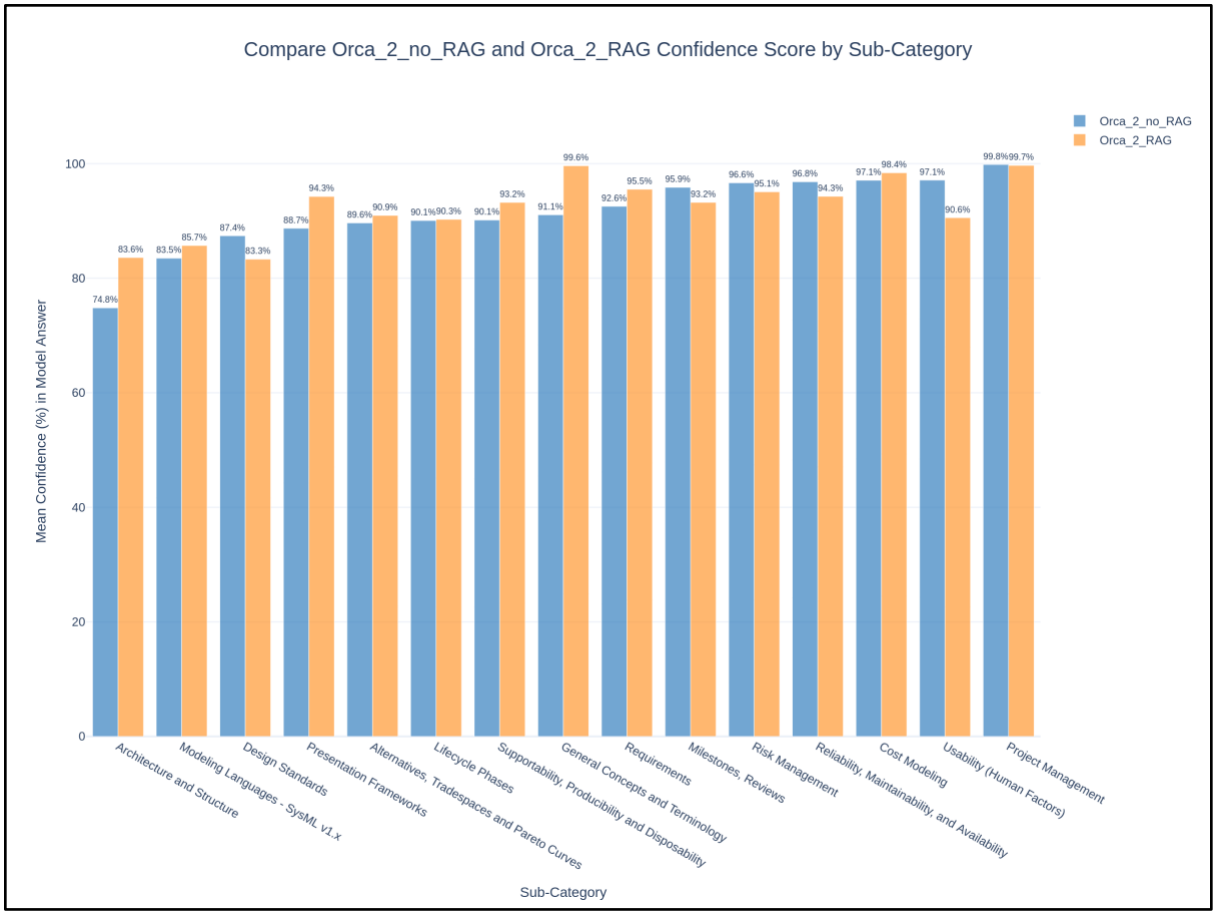
Orca 2 also improved its accuracy using the RAG system, we were able to reduce the error rate on the first question set by 17.65% and by 18.14% on the second. The highest scores were achieved using the current corpus without using LLM as a judge and with only the top 1 retrieved text from the RAG, we believe this model is sensitive to longer prompts but performs well when provided with concise and relevant information, this can be seen in the custom corpus section where we provide the model with just that and achieved the highest accuracy from all the models so far.

start at question 1 sample results:

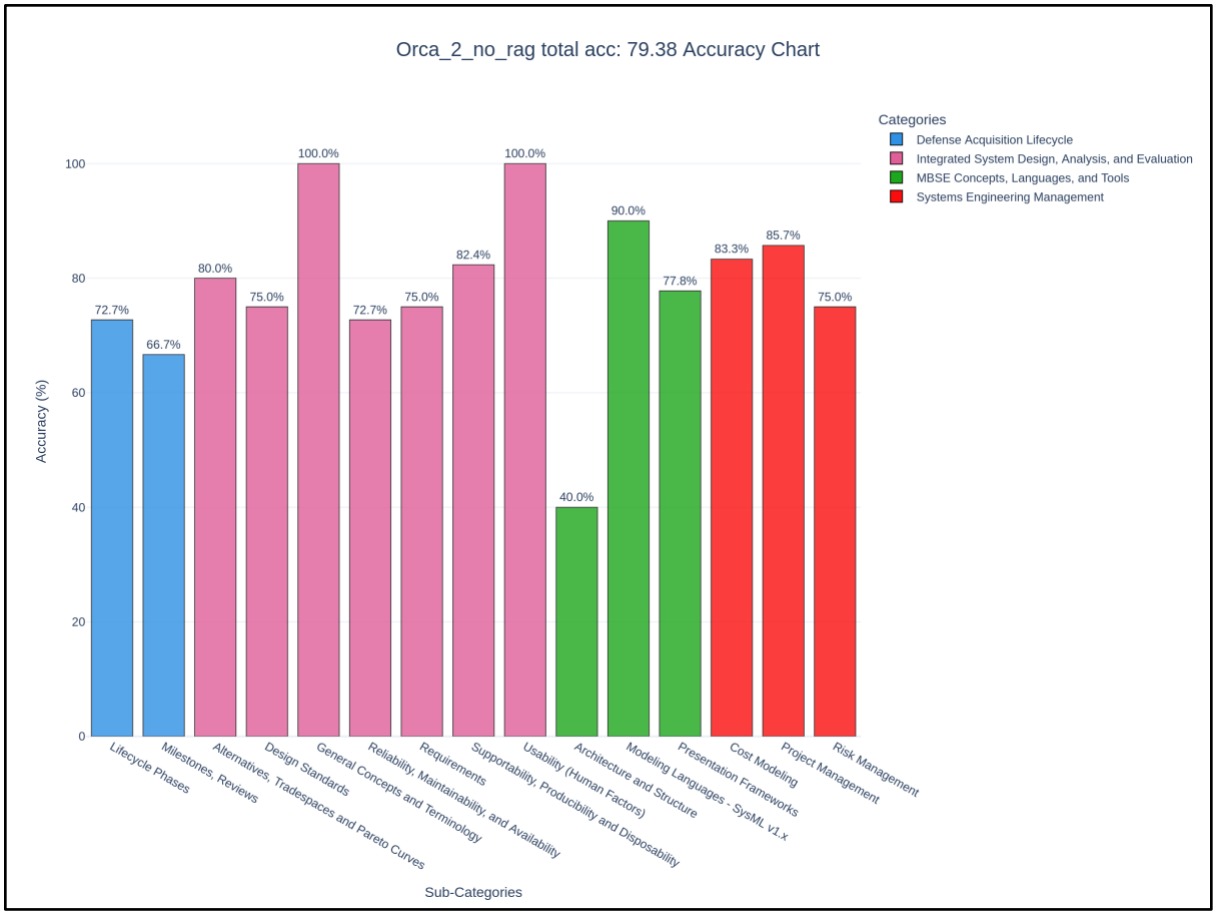


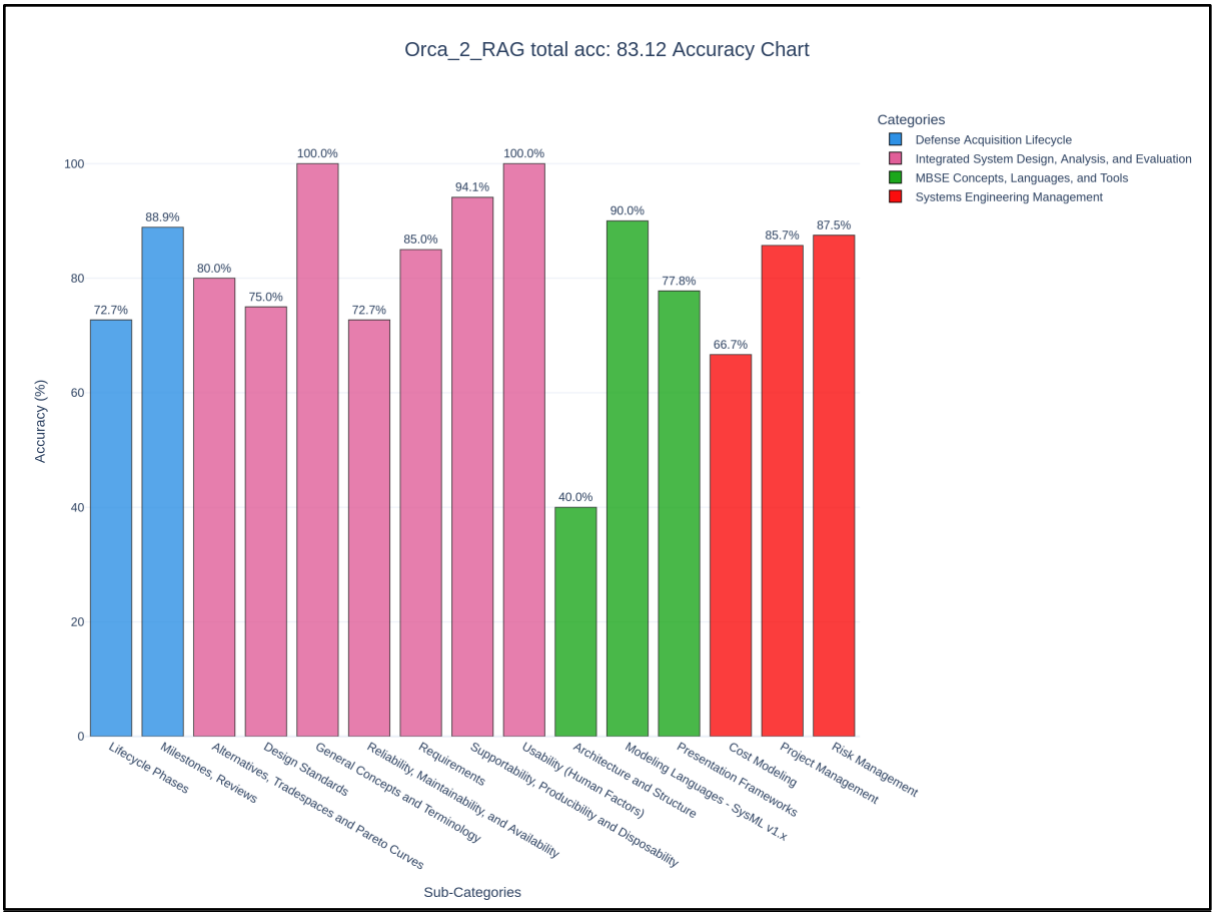

Orca_2_RAG total acc: 82.5 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
Sub-Categories
92.9%
40.0%
80.0%
70.6%
100.0%
81.8%
94.4%
82.4%
100.0%
80.0%
66.7%
88.9%
80.0%
100.0%
75.0%
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management

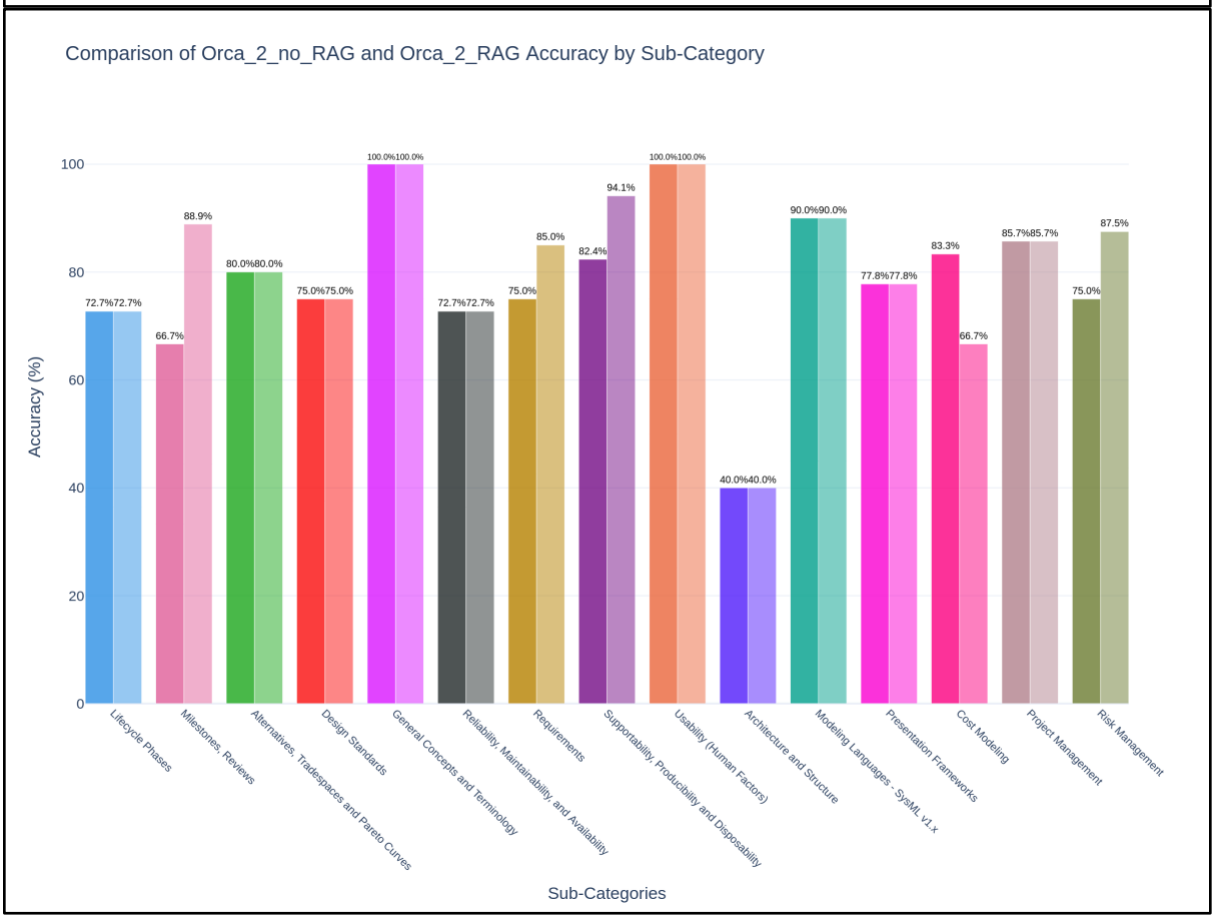

Comparison of Orca_2_no_RAG and Orca_2_RAG Accuracy by Sub-Category
Accuracy (%)
Sub-Categories
71.4%
92.9%
40.0%40.0%
80.0%80.0%
70.6%70.6%
100.0%100.0%
81.8%81.8%
94.4%94.4%
82.4%82.4%
91.7%
100.0%
60.0%
80.0%
66.7%66.7%
77.8%
88.9%
80.0%80.0%
100.0%100.0%
75.0%75.0%
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management

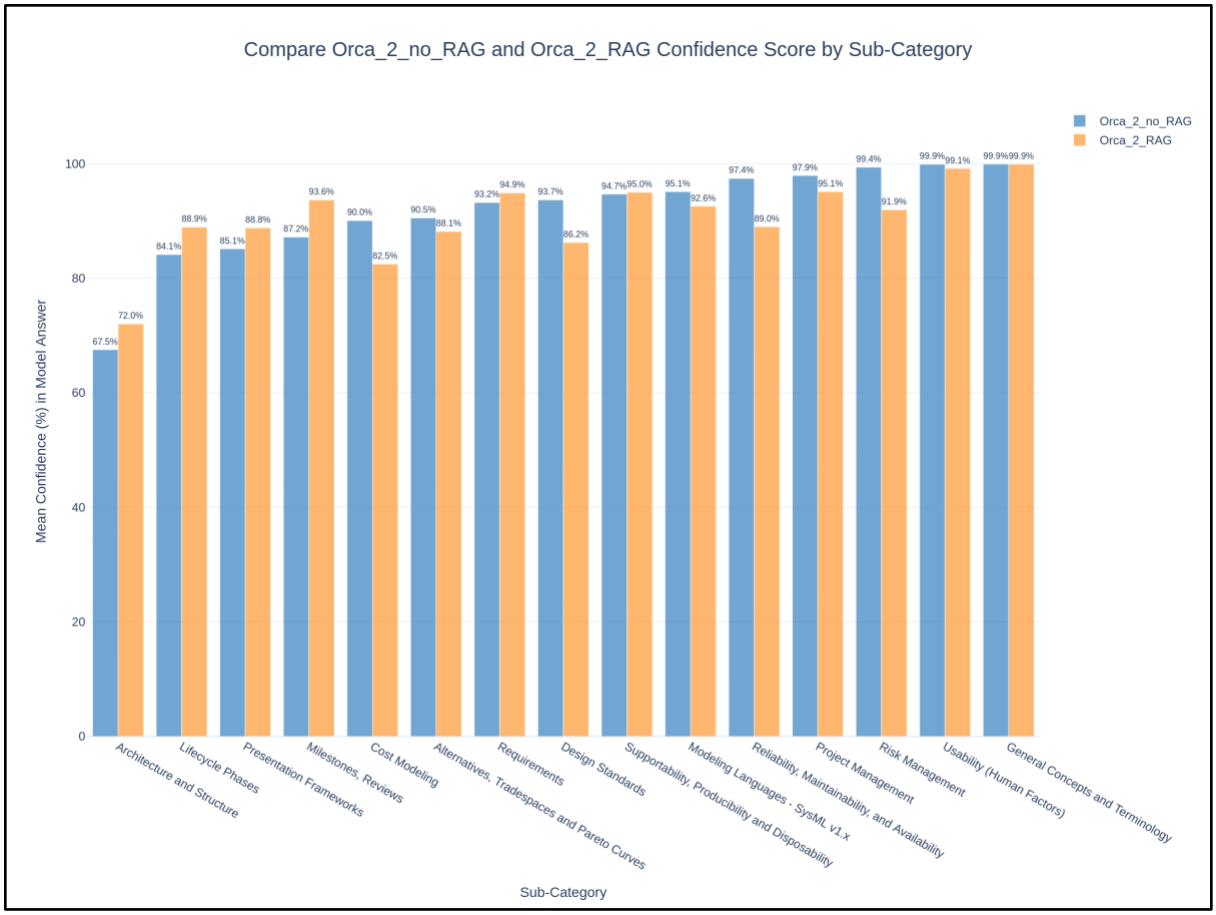


start at question 5 sample results:


Orca_2_no_rag total acc: 79.38 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
Sub-Categories
72.7%
66.7%
80.0%
75.0%
100.0%
72.7%
75.0%
82.4%
100.0%
40.0%
90.0%
77.8%
83.3%
85.7%
75.0%
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management


Orca_2_RAG total acc: 83.12 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
Sub-Categories
72.7%
88.9%
80.0%
75.0%
100.0%
72.7%
85.0%
94.1%
100.0%
40.0%
90.0%
77.8%
66.7%
85.7%
87.5%
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management

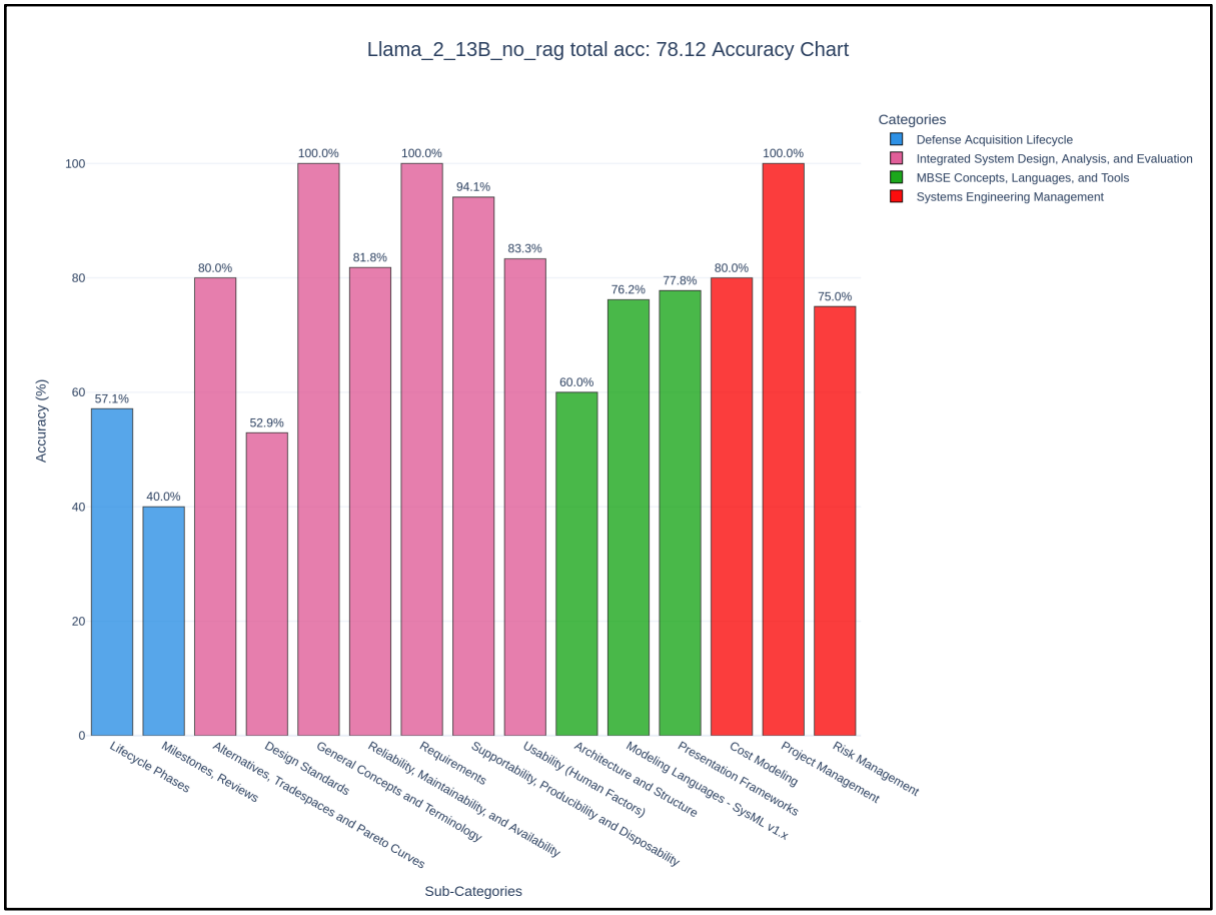

Comparison of Orca_2_no_RAG and Orca_2_RAG Accuracy by Sub-Category
Accuracy (%)
Sub-Categories
72.7%72.7%
66.7%
88.9%
80.0%80.0%
75.0%75.0%
100.0%100.0%
72.7%72.7%
75.0%
85.0%
82.4%
94.1%
100.0%100.0%
40.0%40.0%
90.0%90.0%
77.8%77.8%
83.3%
66.7%
85.7%85.7%
75.0%
87.5%
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management

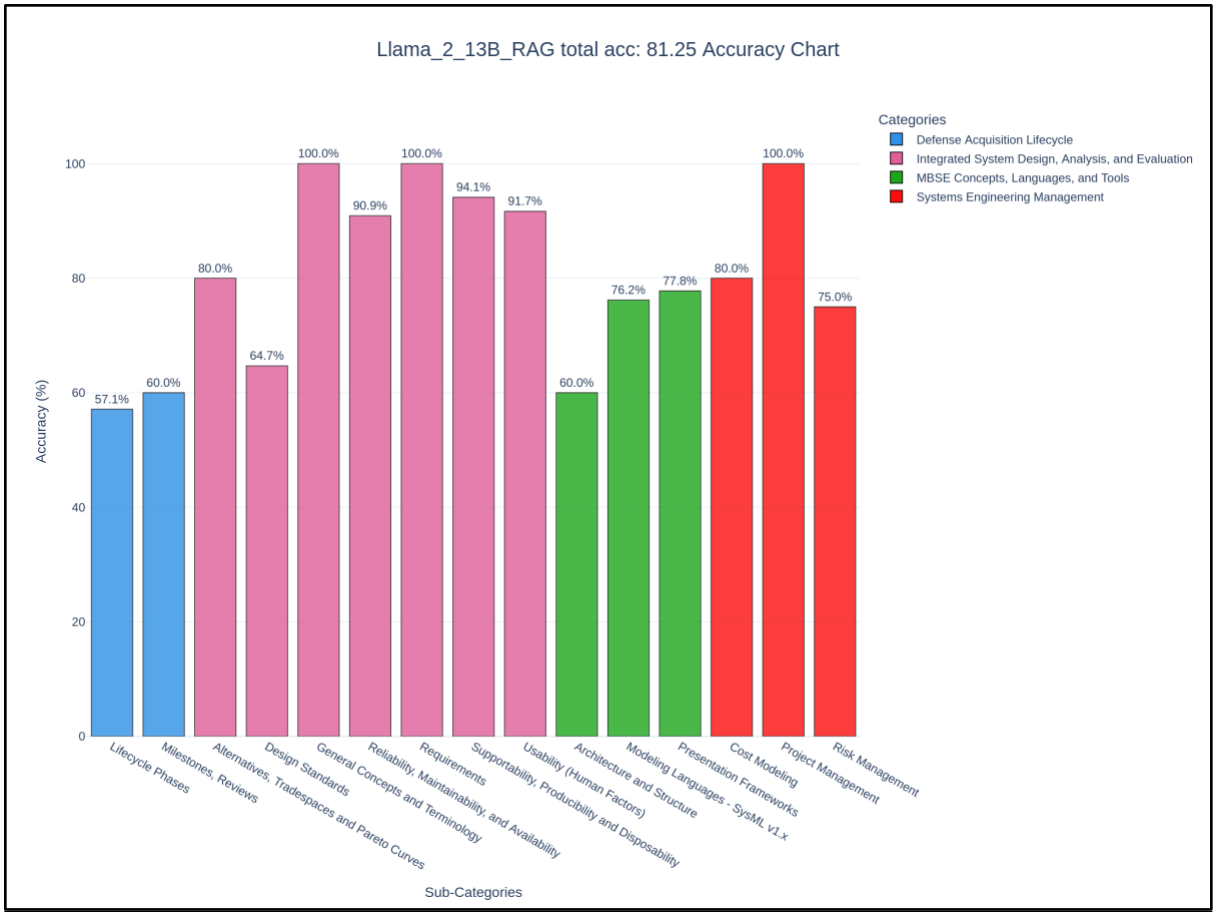

Compare Orca_2_no_RAG and Orca_2_RAG Confidence Score by Sub-Category
Orca_2_no_RAG
Orca_2_RAG
Mean Confidence (%) in Model Answer
Sub-Category
0
20
40
60
80
100
67.5%
72.0%
84.1%
88.9%
85.1%
88.8%
87.2%
93.6%
90.0%
82.5%
90.5%
88.1%
93.2%
94.9%
93.7%
86.2%
94.7%
95.0%
95.1%
92.6%
97.4%
89.0%
97.9%
95.1%
99.4%
91.9%
99.9%
99.1%
99.9%
99.9%
Architecture and Structure
Lifecycle Phases
Presentation Frameworks
Milestones, Reviews
Cost Modeling
Alternatives, Tradespaces and Pareto Curves
Requirements
Design Standards
Supportability, Producibility and Disposability
Modeling Languages - SysML v1.x
Reliability, Maintainability, and Availability
Project Management
Risk Management
Usability (Human Factors)
General Concepts and Terminology

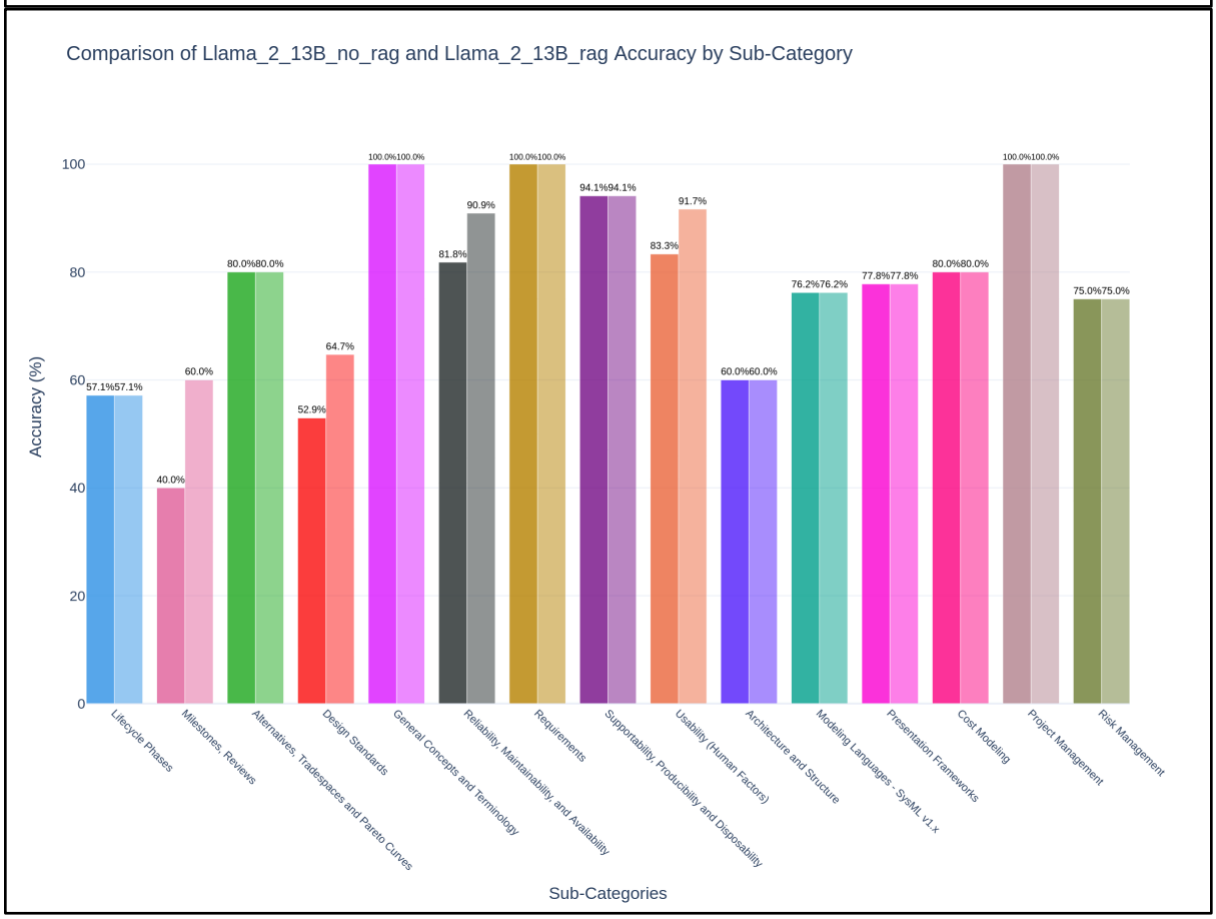
Llama 2 13B

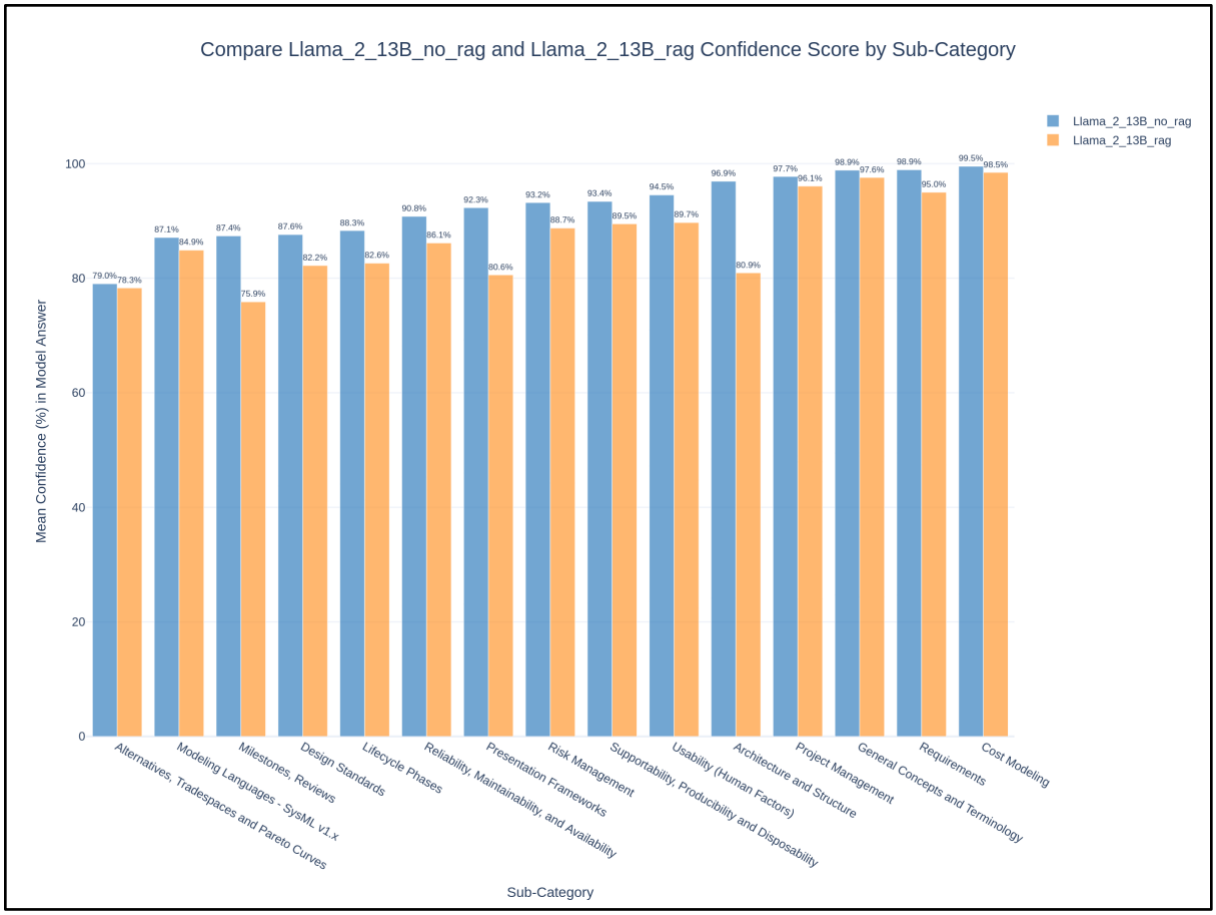
Llama 2 showed similar improvement to Orca 2 with reduction in error rates of 14.31% for the first question set and 9.07% for the second. Similarly, the highest scores were achieved using the current corpus without using LLM as a judge and with the top 1 retrieved text from the RAG, we believe that like Orca this model is sensitive to longer prompts but performs well when provided with concise and relevant information, and again we achieved drastic improvement with the custom corpus.

start at question 1 sample results:

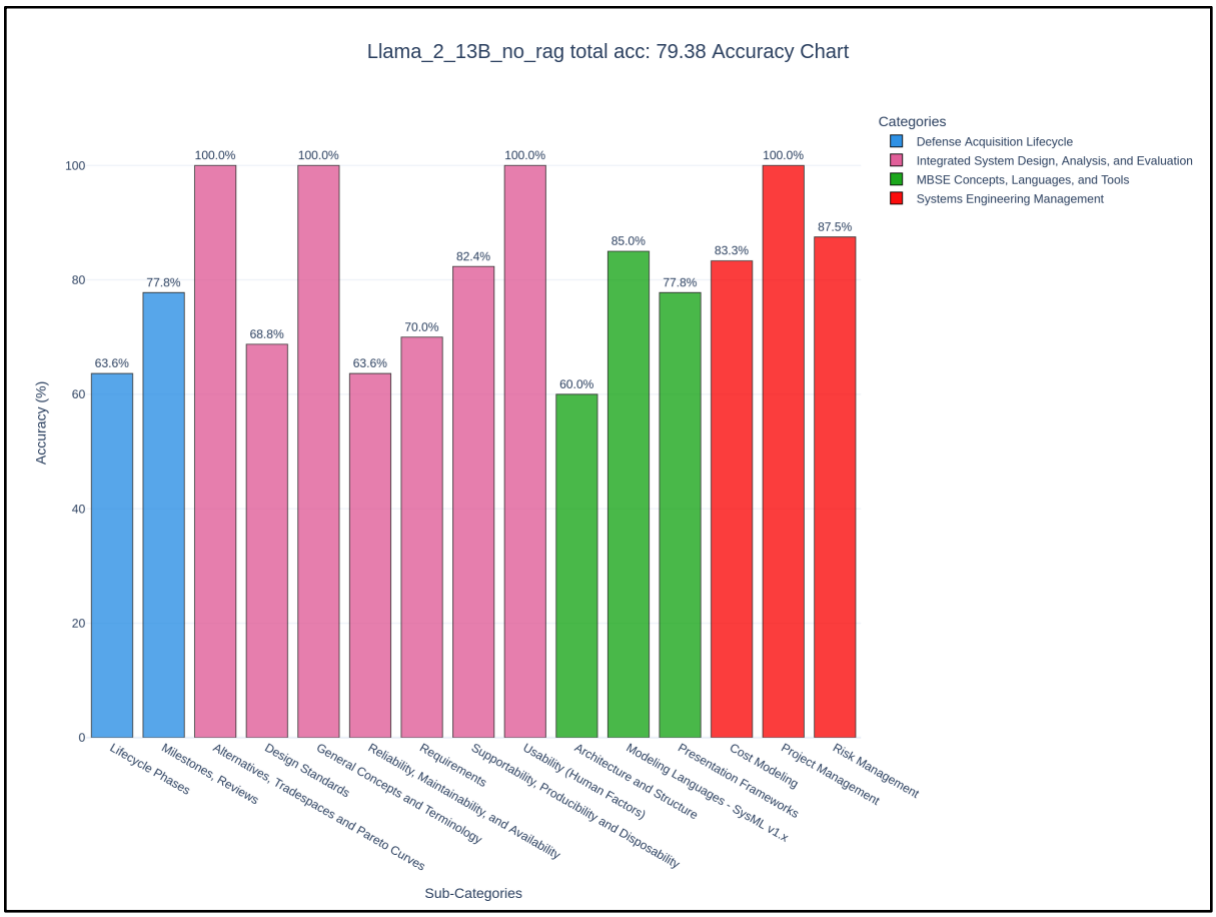


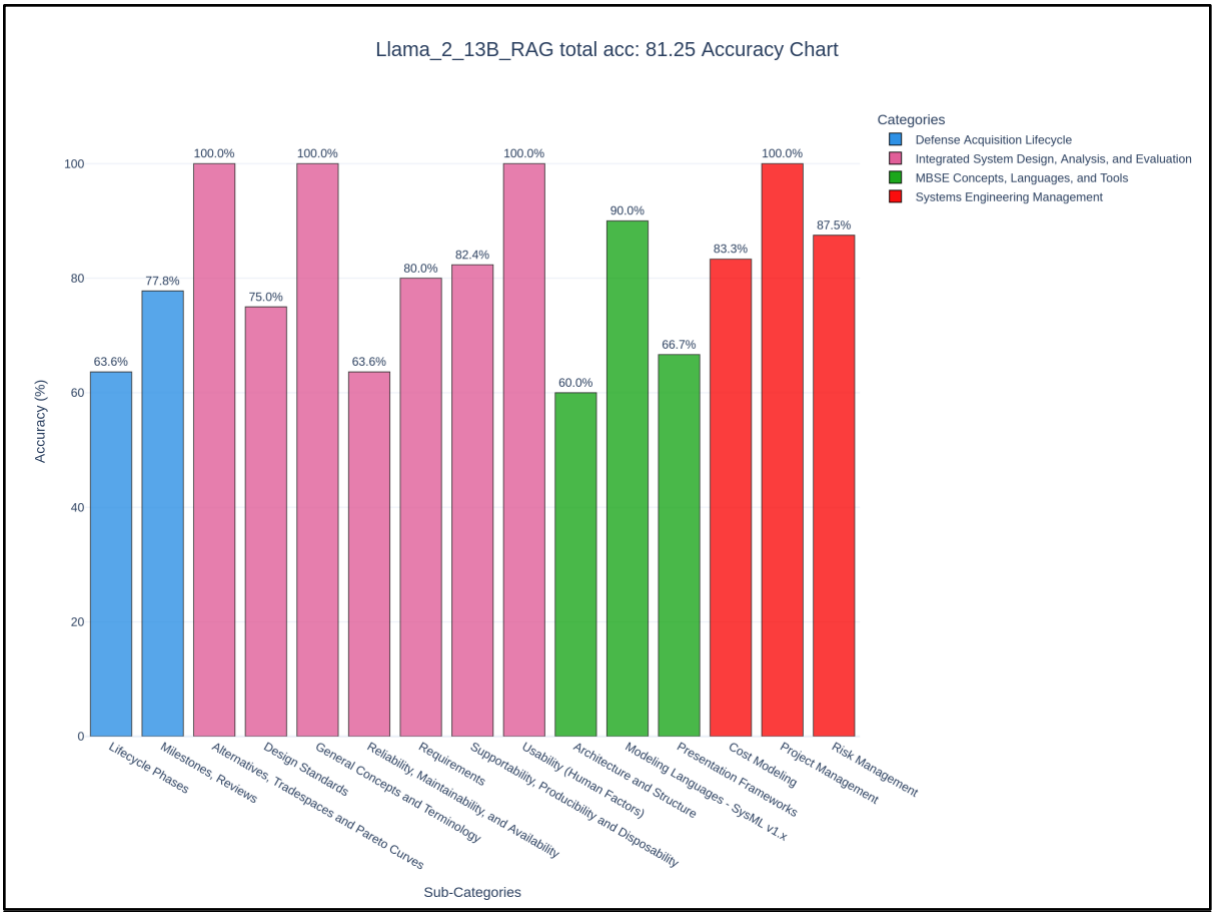

Llama_2_13B_RAG total acc: 81.25 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
Sub-Categories
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management
57.1%
60.0%
80.0%
64.7%
100.0%
90.9%
100.0%
94.1%
91.7%
60.0%
76.2%
77.8%
80.0%
100.0%
75.0%

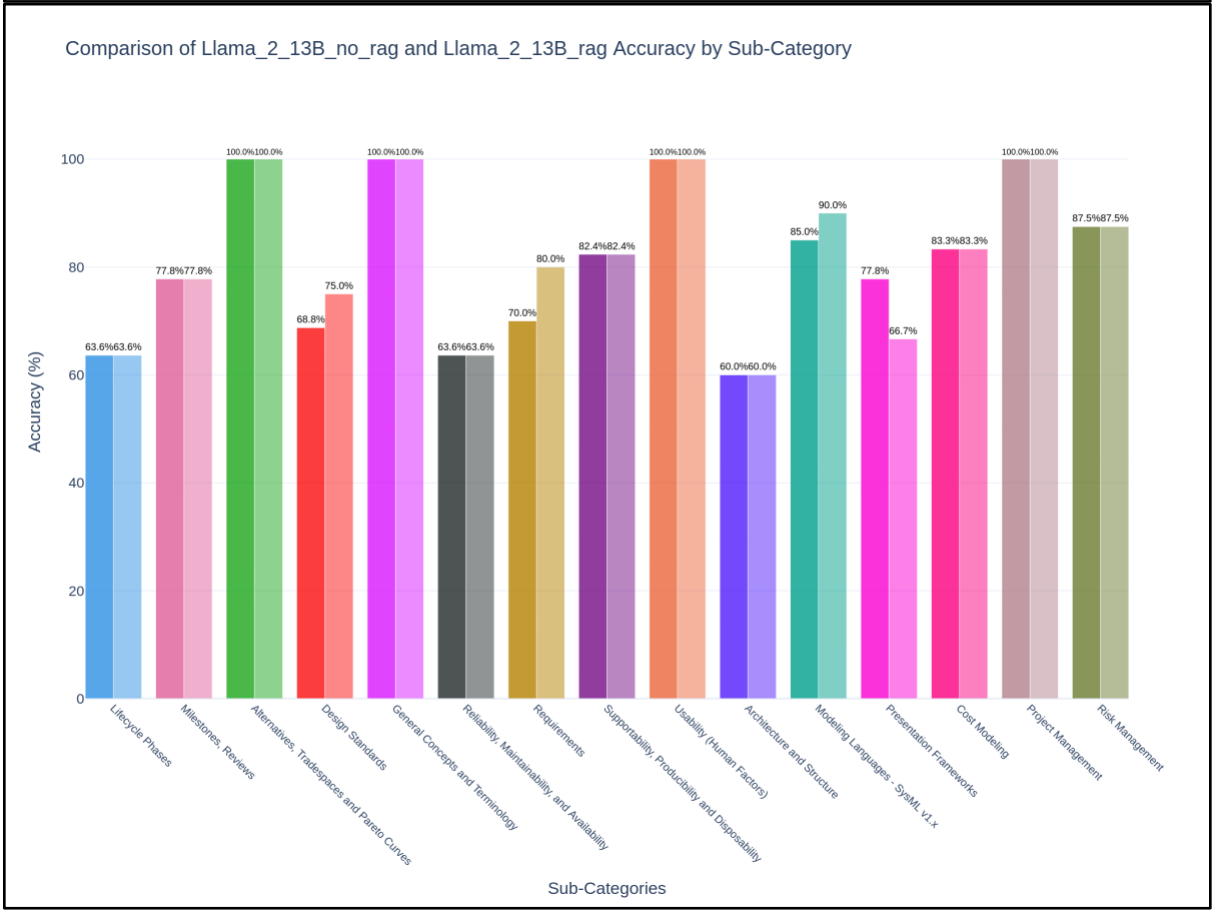

Comparison of Llama_2_13B_no_rag and Llama_2_13B_rag Accuracy by Sub-Category
Accuracy (%)
Sub-Categories
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management

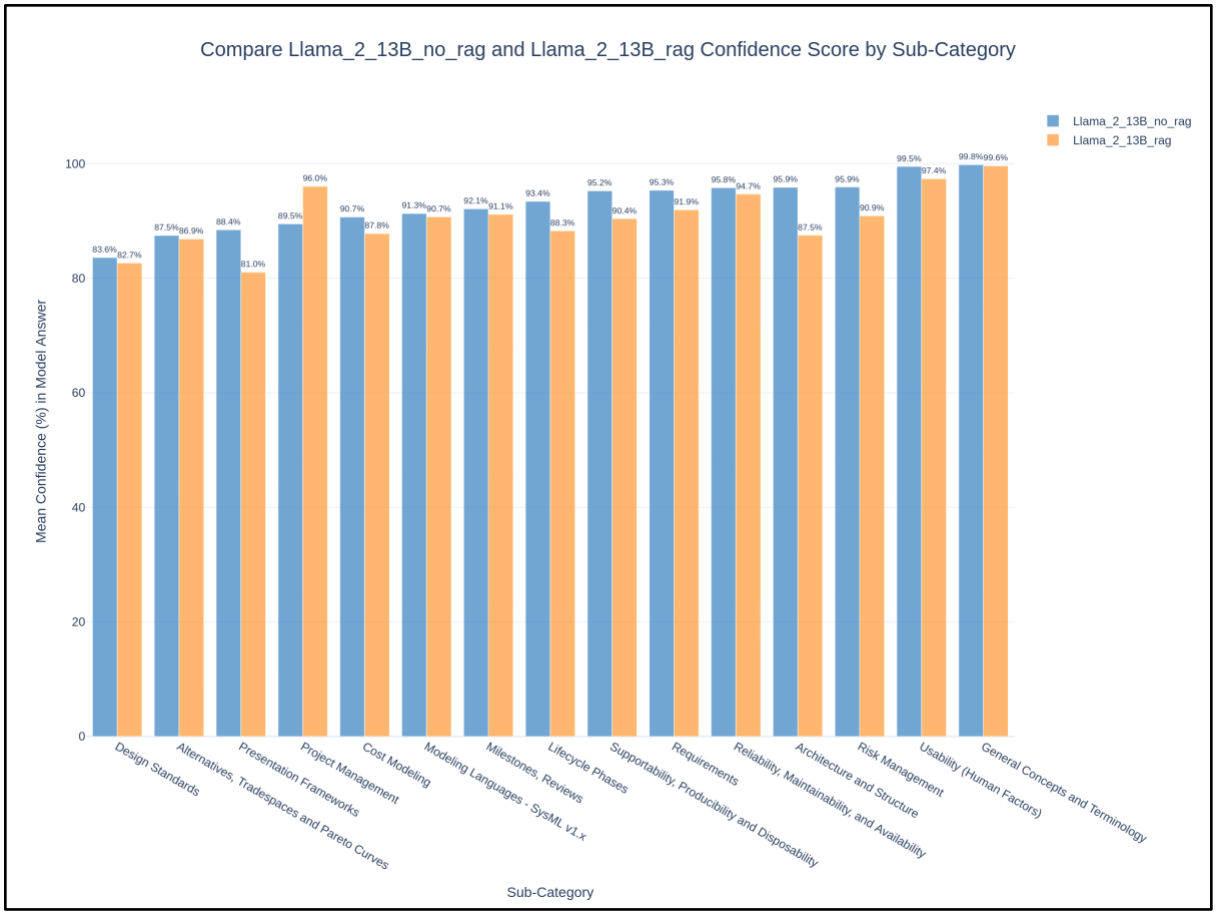

Compare Llama_2_13B_no_rag and Llama_2_13B_rag Confidence Score by Sub-Category
Llama_2_13B_no_rag
Llama_2_13B_rag
Mean Confidence (%) in Model Answer
Sub-Category
0
20
40
60
80
100
79.0%
78.3%
87.1%
84.9%
87.4%
75.9%
87.6%
82.2%
88.3%
82.6%
90.8%
86.1%
92.3%
80.6%
93.2%
88.7%
93.4%
89.5%
94.5%
89.7%
96.9%
80.9%
97.7%
96.1%
98.9%
97.6%
98.9%
95.0%
99.5%
98.5%
Alternatives, Tradespaces and Pareto Curves
Modeling Languages - SysML v1.x
Milestones, Reviews
Design Standards
Lifecycle Phases
Reliability, Maintainability, and Availability
Presentation Frameworks
Risk Management
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Project Management
General Concepts and Terminology
Requirements
Cost Modeling

start at question 5 sample results:




Llama_2_13B_RAG total acc: 81.25 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
Sub-Categories
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management
63.6%
77.8%
100.0%
75.0%
100.0%
63.6%
80.0%
82.4%
100.0%
60.0%
90.0%
66.7%
83.3%
100.0%
87.5%


Comparison of Llama_2_13B_no_rag and Llama_2_13B_rag Accuracy by Sub-Category
Accuracy (%)
Sub-Categories
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management


Compare Llama_2_13B_no_rag and Llama_2_13B_rag Confidence Score by Sub-Category
Llama_2_13B_no_rag
Llama_2_13B_rag
Mean Confidence (%) in Model Answer
0
20
40
60
80
100
83.6%
82.7%
87.5%
86.9%
88.4%
81.0%
89.5%
96.0%
90.7%
87.8%
91.3%
90.7%
92.1%
91.1%
93.4%
88.3%
95.2%
90.4%
95.3%
91.9%
95.8%
94.7%
95.9%
87.5%
95.9%
90.9%
99.5%
97.4%
99.8%
99.6%
Design Standards
Alternatives, Tradespaces and Pareto Curves
Presentation Frameworks
Project Management
Cost Modeling
Modeling Languages - SysML V1.x
Milestones, Reviews
Lifecycle Phases
Supportability, Producibility and Disposability
Requirements
Reliability, Maintainability, and Availability
Architecture and Structure
Risk Management
Usability (Human Factors)
General Concepts and Terminology
Sub-Category

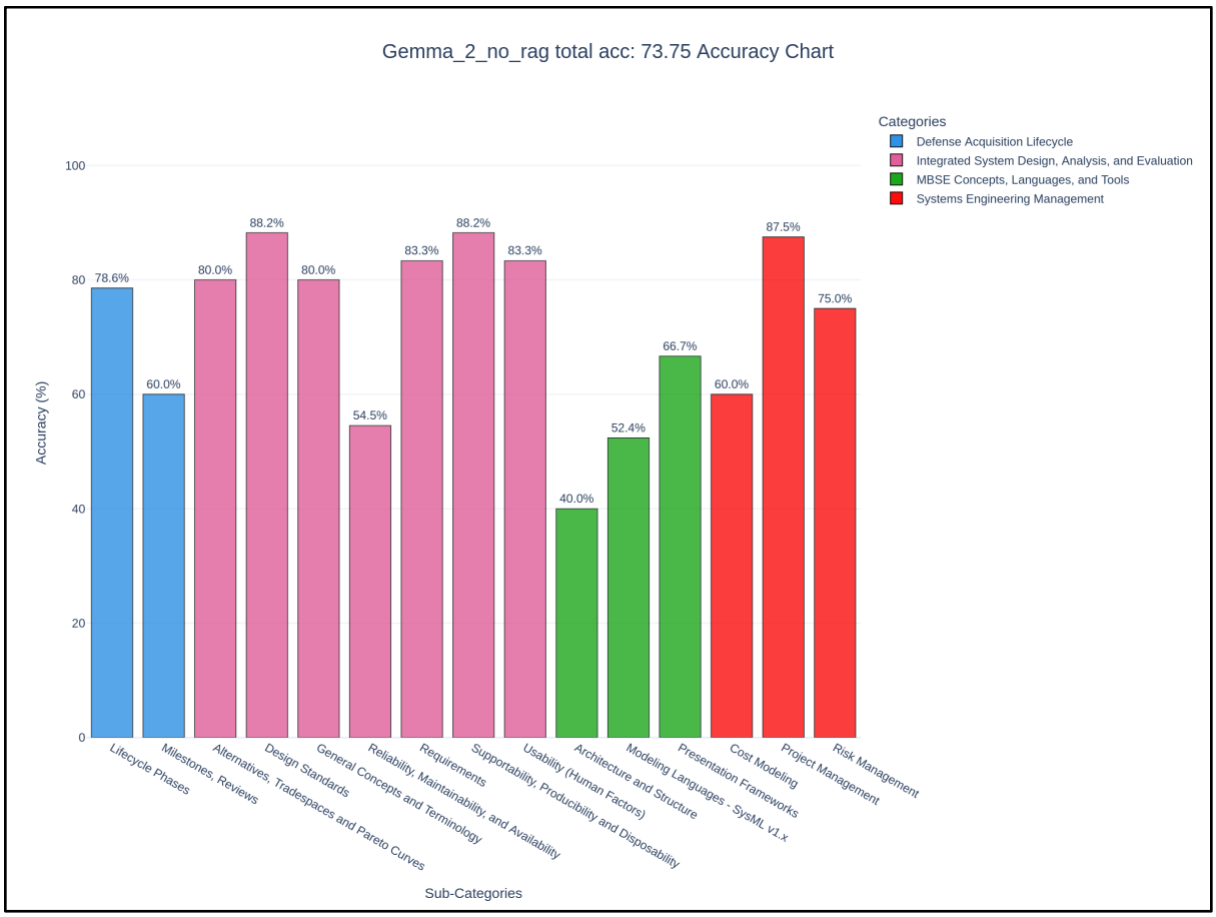
Gemma 2 2B

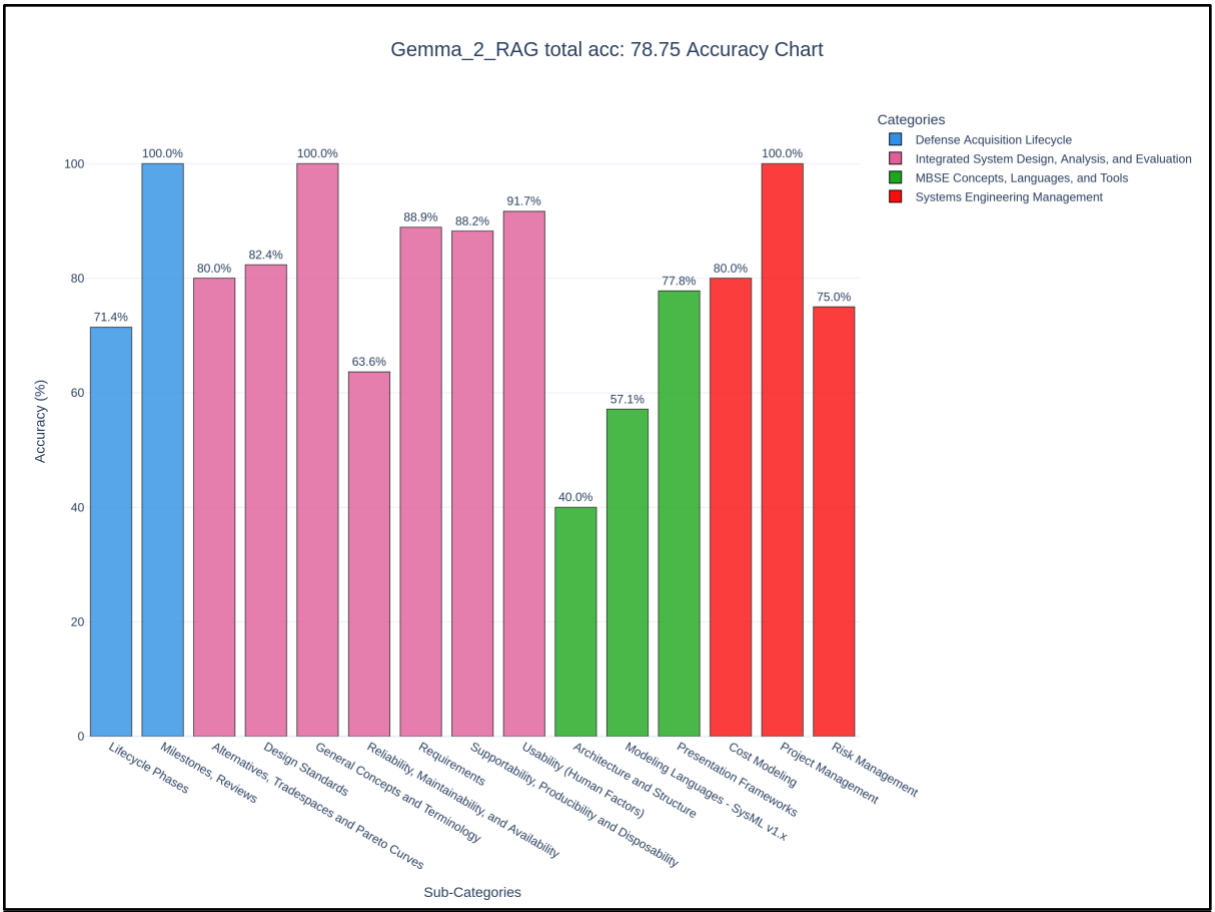
Gemma 2 demonstrated significant improvement, with accuracy increasing from 73.75% to 78.75% on the first question set, corresponding to a 19.05% reduction in error rate, and from 63.75% to 75.0% on the second question set, corresponding to a 31.03% reduction in error rate. The best results were achieved using LLM as a judge and retrieving only the top 1 relevant document. On the custom corpus, Gemma 2 reached an accuracy of 80%, showcasing its ability to handle concise and highly relevant information effectively.

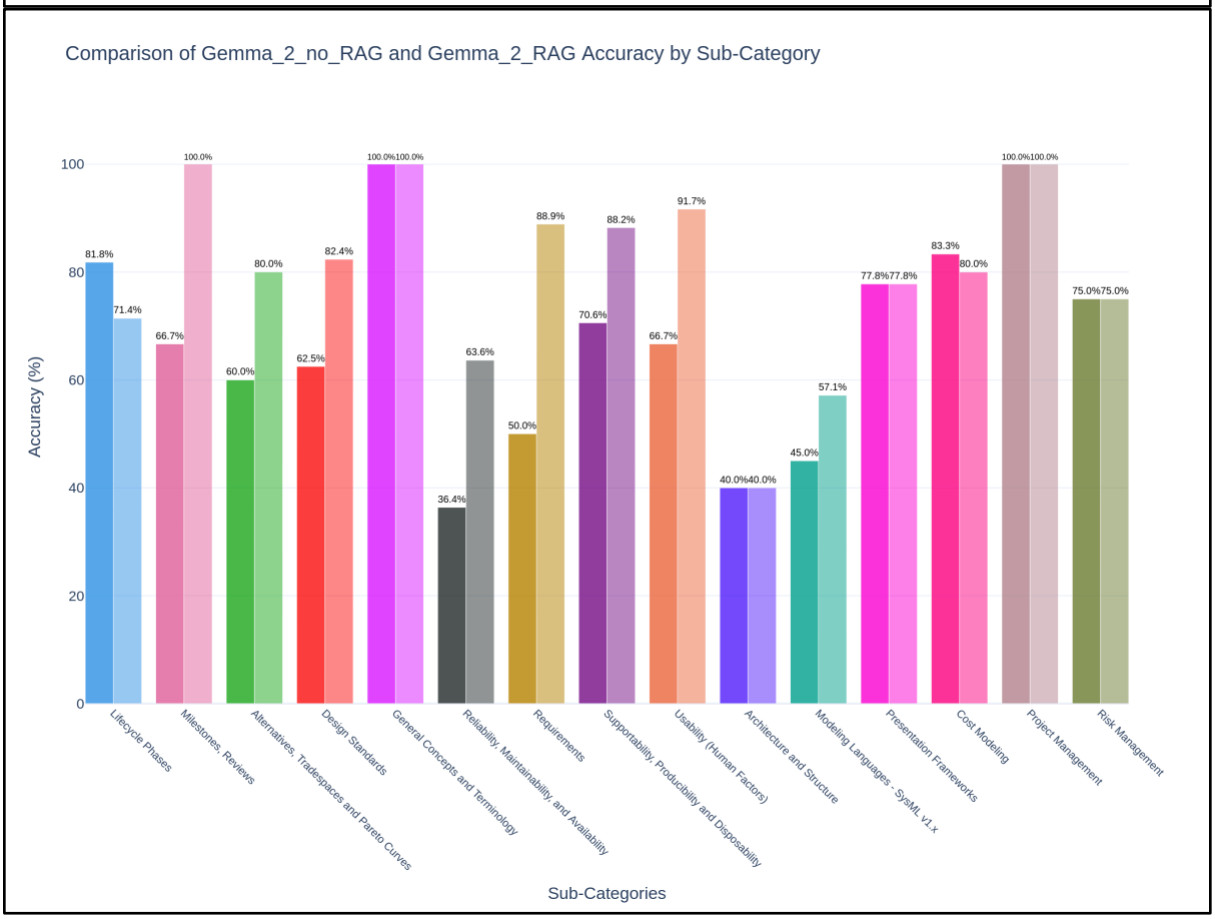
This performance highlights the potential of smaller models to achieve results comparable to those of much larger architectures, such as Llama 2 13B, when paired with an optimized RAG system. By combining techniques like targeted retrieval, the use of an LLM judge for filtering, and concise input formatting, the limitations of smaller parameter counts and simpler architectures can be effectively mitigated. These results suggest that improvements in retrieval and data handling can close the gap between small and large models, showcasing the importance of retrieval-augmented techniques in performance optimization.

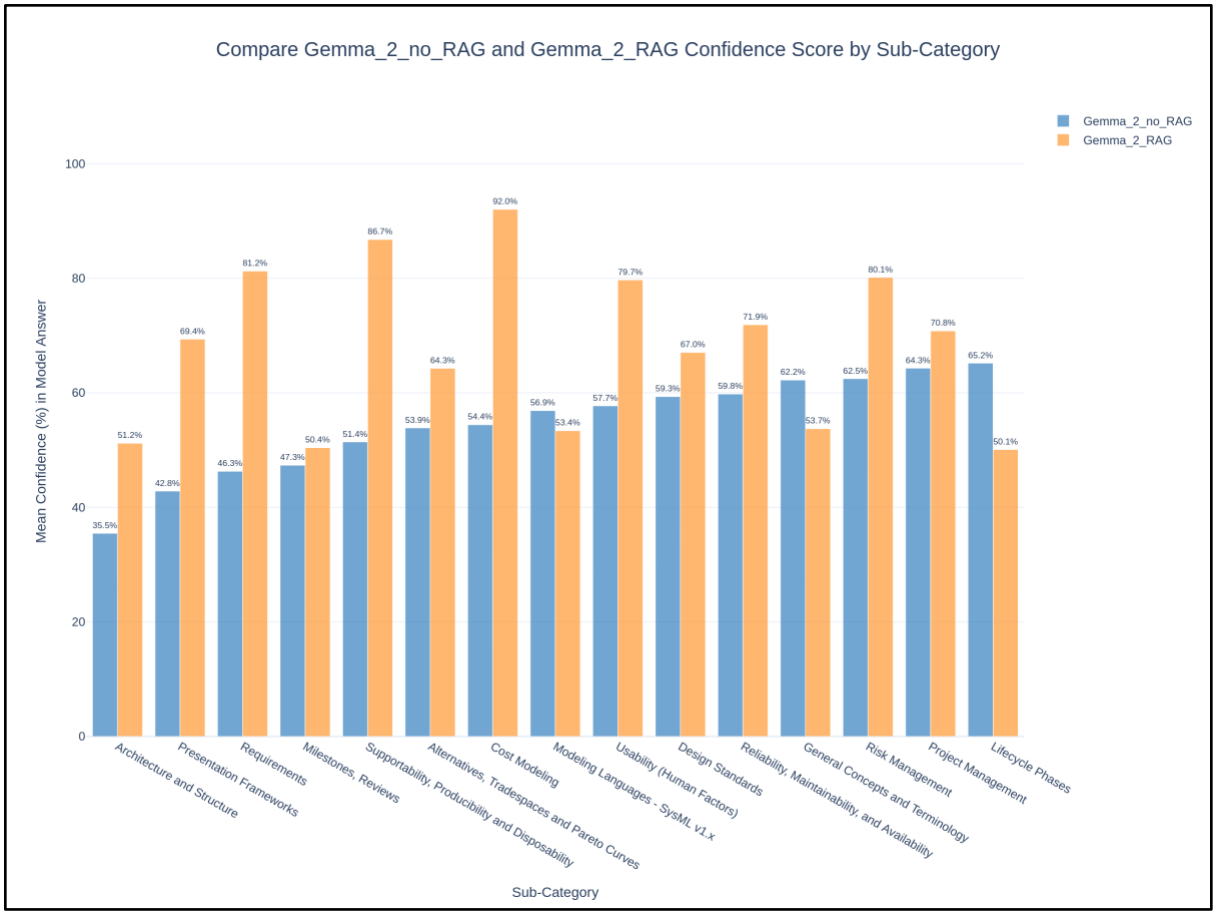
This is especially important for environments with limited computational resources, where smaller models like Gemma 2 can be used more cost-effectively than larger ones. Additionally, achieving high accuracy with smaller models addresses the increasing demand for energy-efficient and sustainable technology. By prioritizing retrieval methods and input optimization over simply increasing model size, these findings demonstrate a practical approach to making high-performing systems more accessible, particularly in specialized fields such as Systems Engineering.

start at question 1 sample results:

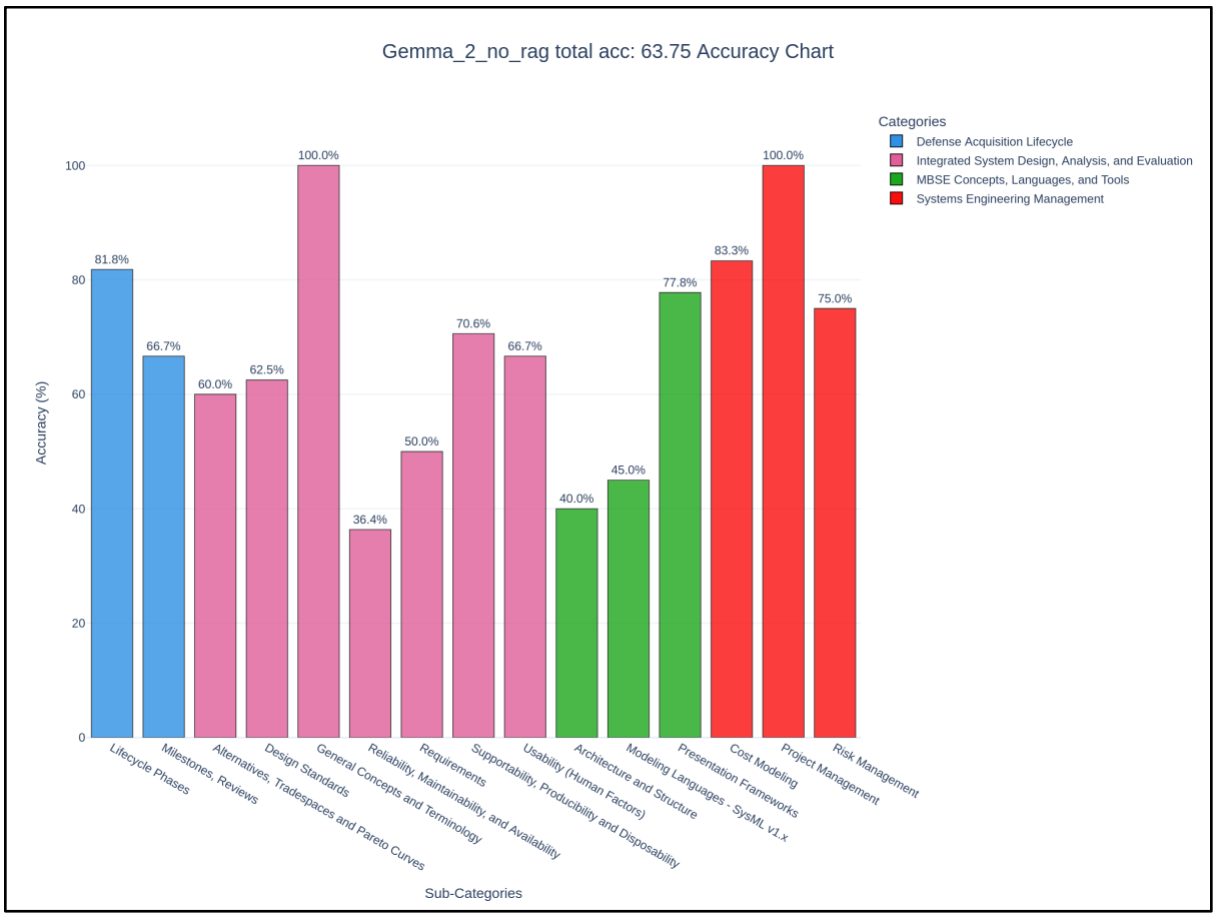


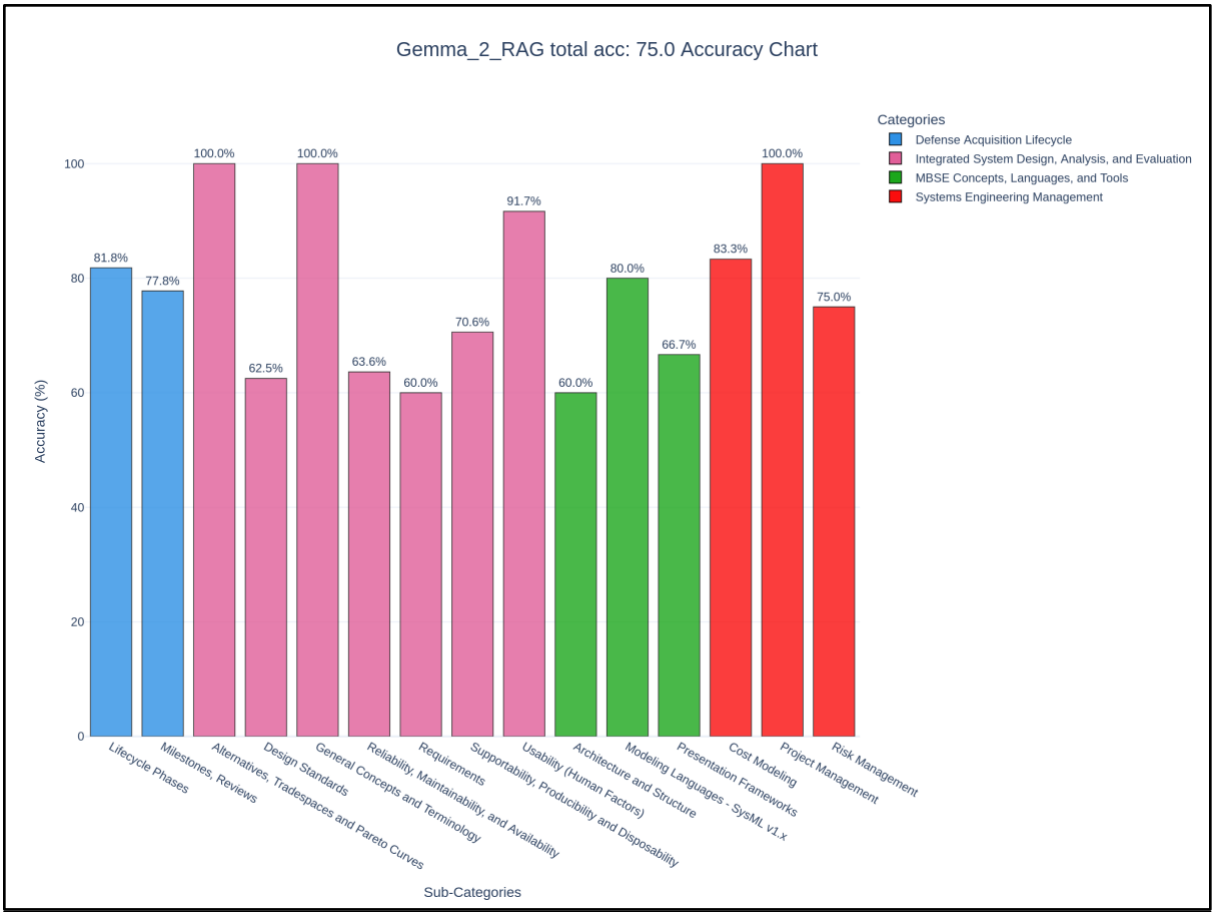

Gemma_2_RAG total acc: 78.75 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
Sub-Categories
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management

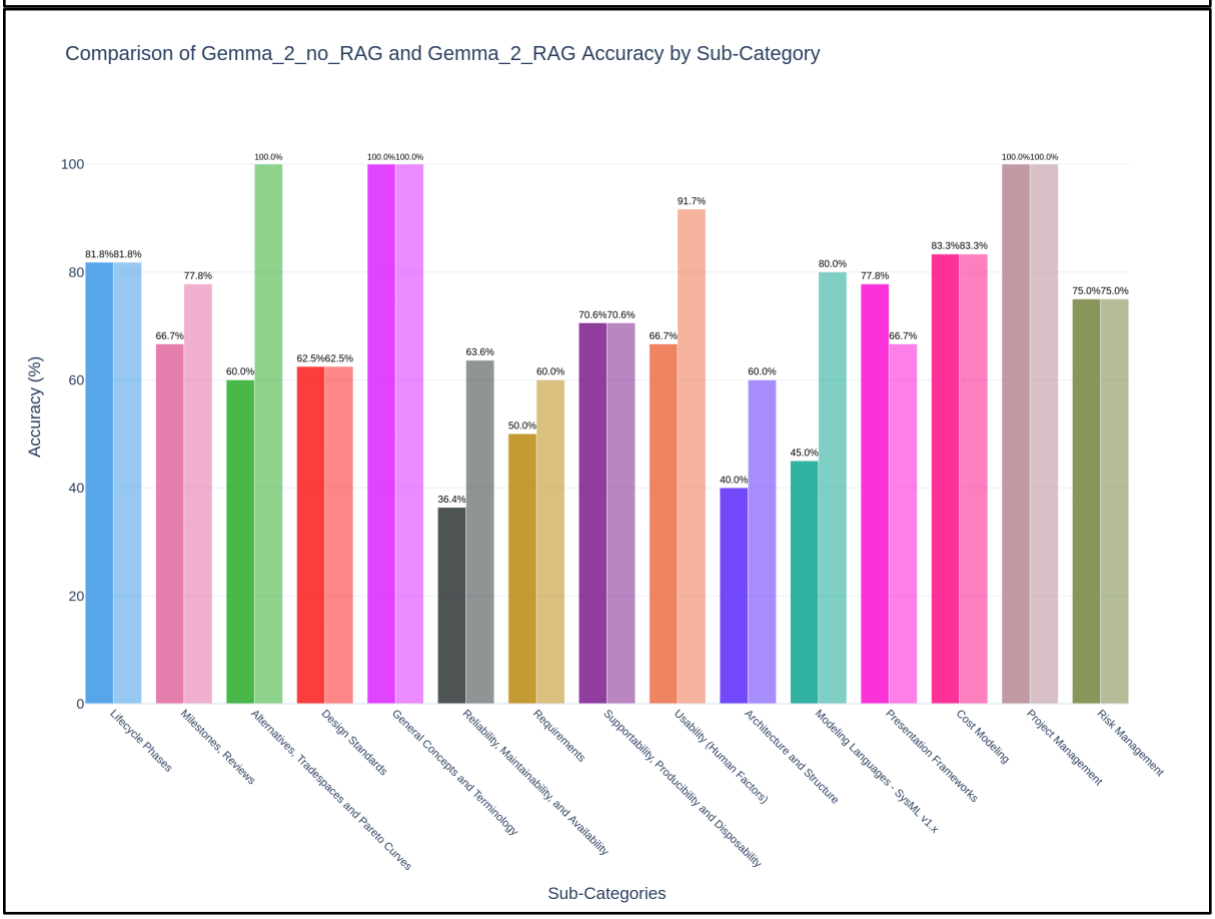

Comparison of Gemma_2_no_RAG and Gemma_2_RAG Accuracy by Sub-Category
Accuracy (%)
Sub-Categories
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management

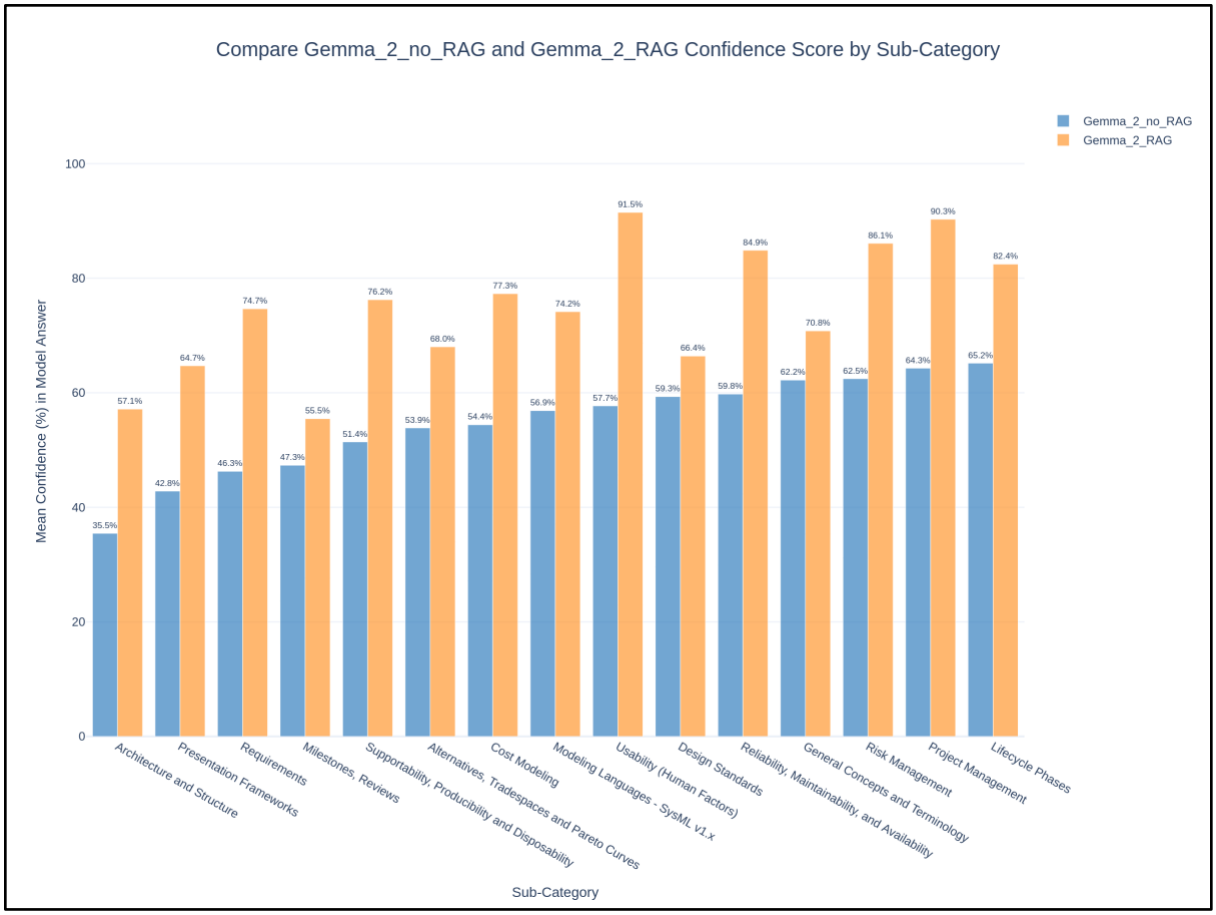

Compare Gemma_2_no_RAG and Gemma_2_RAG Confidence Score by Sub-Category
Gemma_2_no_RAG
Gemma_2_RAG
Mean Confidence (%) in Model Answer
Sub-Category
0
20
40
60
80
100
35.5%
51.2%
42.8%
69.4%
46.3%
81.2%
47.3%
50.4%
51.4%
86.7%
53.9%
64.3%
54.4%
92.0%
56.9%
53.4%
57.7%
79.7%
59.3%
67.0%
59.8%
71.9%
62.2%
53.7%
62.5%
80.1%
64.3%
70.8%
65.2%
50.1%
Architecture and Structure
Presentation Frameworks
Requirements
Milestones, Reviews
Supportability, Producibility and Disposability
Alternatives, Tradespaces and Pareto Curves
Cost Modeling
Modeling Languages - SysML v1.x
Usability (Human Factors)
Design Standards
Reliability, Maintainability, and Availability
General Concepts and Terminology
Risk Management
Project Management
Lifecycle Phases

start at question 5 sample results:




Gemma_2_RAG total acc: 75.0 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
Sub-Categories
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management
81.8%
77.8%
100.0%
62.5%
100.0%
63.6%
60.0%
70.6%
91.7%
60.0%
80.0%
66.7%
83.3%
100.0%
75.0%


Comparison of Gemma_2_no_RAG and Gemma_2_RAG Accuracy by Sub-Category
Accuracy (%)
Sub-Categories
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management


Compare Gemma_2_no_RAG and Gemma_2_RAG Confidence Score by Sub-Category
Gemma_2_no_RAG
Gemma_2_RAG
Mean Confidence (%) in Model Answer
Sub-Category
0
20
40
60
80
100
35.5%
57.1%
42.8%
64.7%
46.3%
74.7%
47.3%
55.5%
51.4%
76.2%
53.9%
68.0%
54.4%
77.3%
56.9%
74.2%
57.7%
91.5%
59.3%
66.4%
59.8%
84.9%
62.2%
70.8%
62.5%
86.1%
64.3%
90.3%
65.2%
82.4%
Architecture and Structure
Presentation Frameworks
Requirements
Milestones, Reviews
Supportability, Producibility and Disposability
Alternatives, Tradespaces and Pareto Curves
Cost Modeling
Modeling Languages - SysML v1.x
Usability (Human Factors)
Design Standards
Reliability, Maintainability, and Availability
General Concepts and Terminology
Risk Management
Project Management
Lifecycle Phases

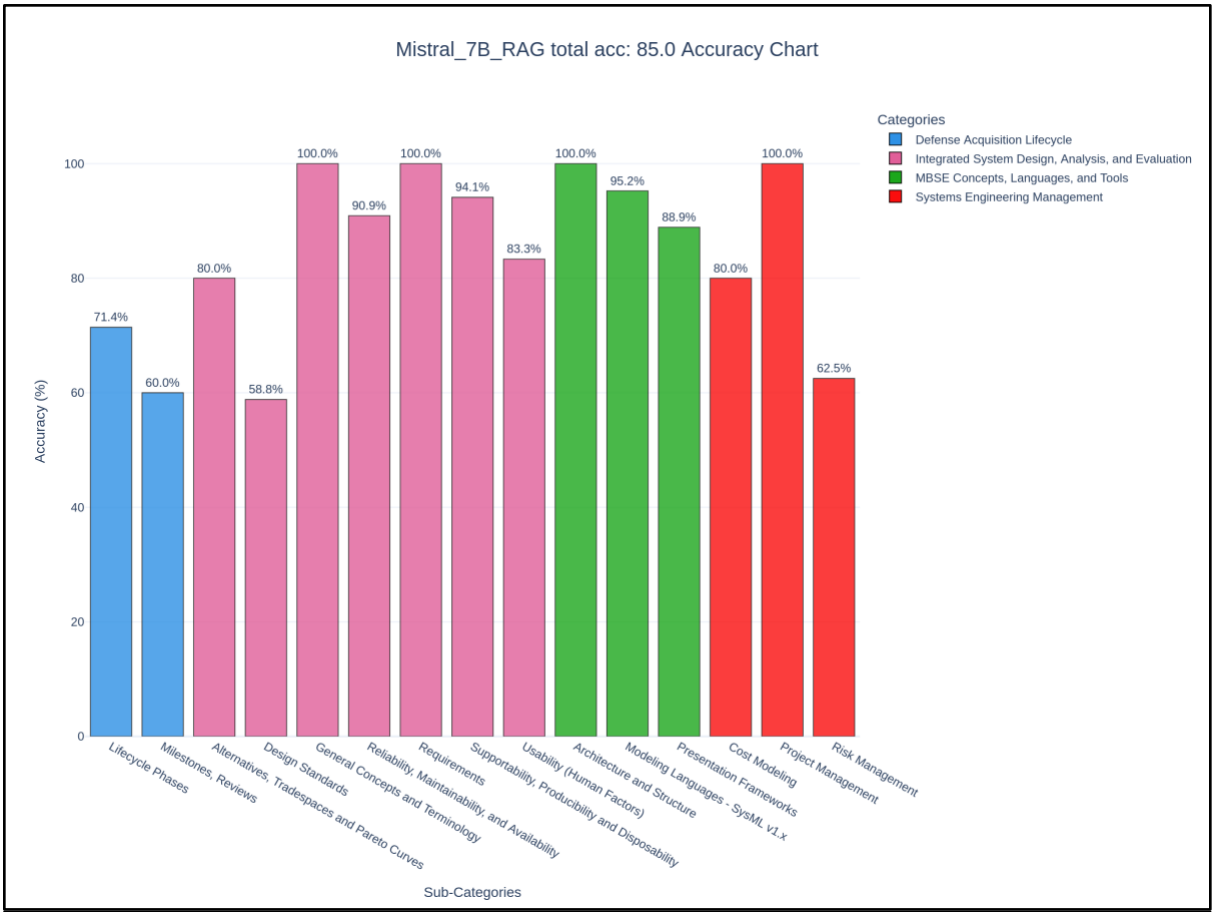
Custom corpus Testing Results

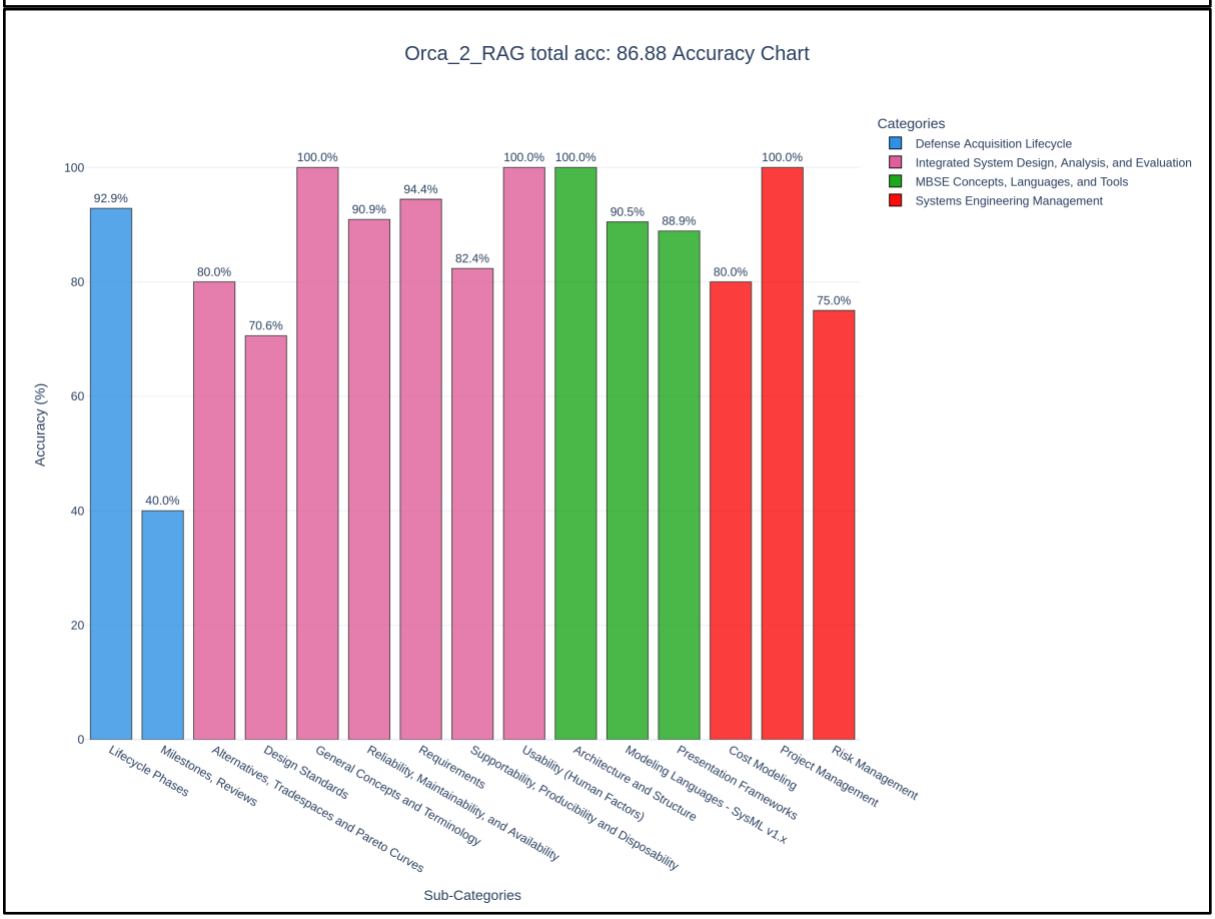
We posed 160 questions, each with multiple-choice options, to GPT-4, instructing it to generate concise and relevant paragraphs of information tailored to answering each question. These generated responses were then concatenated into a single text file, which served as a custom dataset for our Retrieval-Augmented Generation (RAG) system. The primary purpose of this exercise was to conduct a sanity check for our RAG system—essentially, to evaluate its performance when provided with highly relevant information. In simple terms, we aimed to determine whether the system would excel when fed with pre-validated, accurate data.

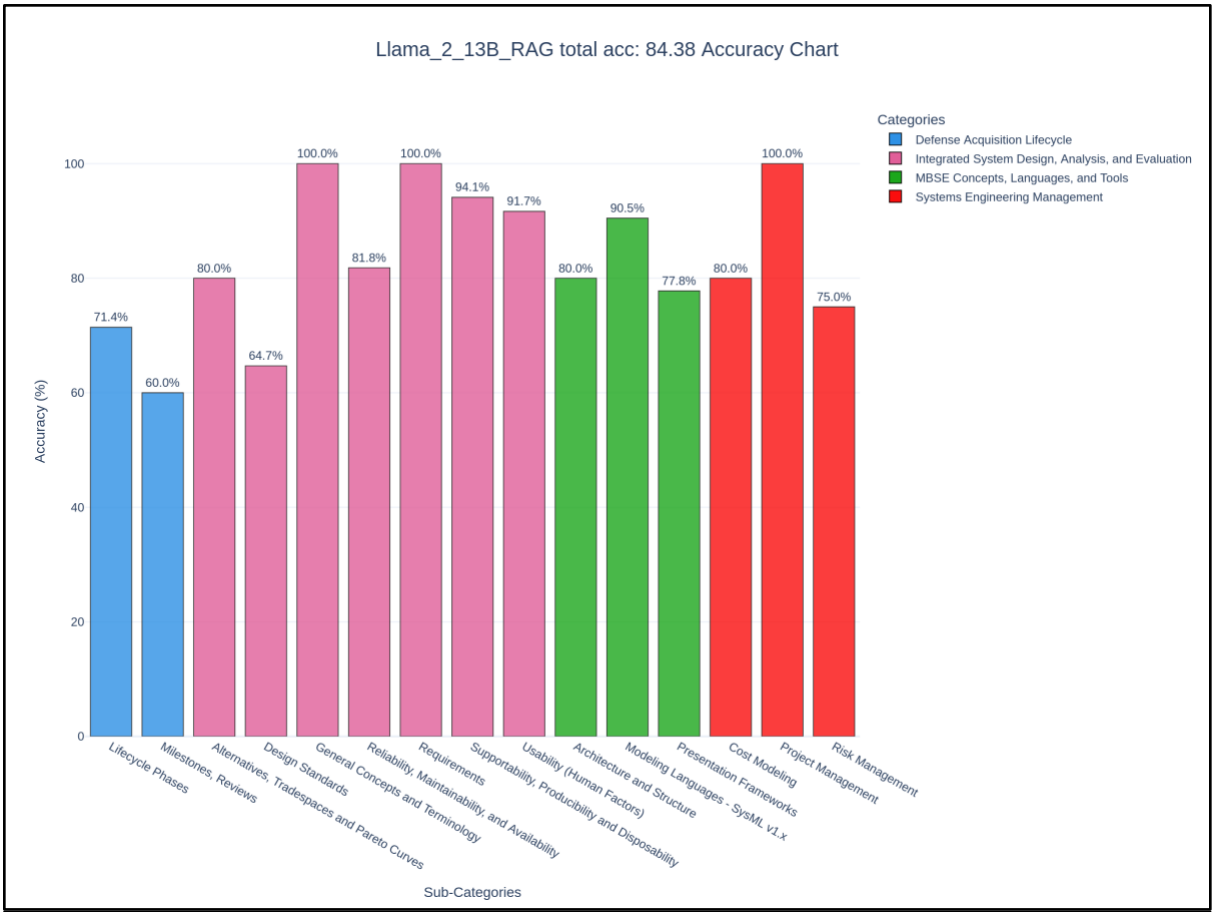
Although GPT-4 is widely regarded as a reliable tool, its outputs in this context were not entirely accurate, and the resulting custom corpus contained minor inconsistencies. This highlighted the importance of clean, high-quality datasets for achieving optimal RAG performance. Furthermore, the experience underscored the potential of using large language models (LLMs) like GPT-4 as a judge for relevance. By validating the contextual relevance of retrieved documents and reducing noise, an LLM can significantly enhance the accuracy and effectiveness of RAG systems.

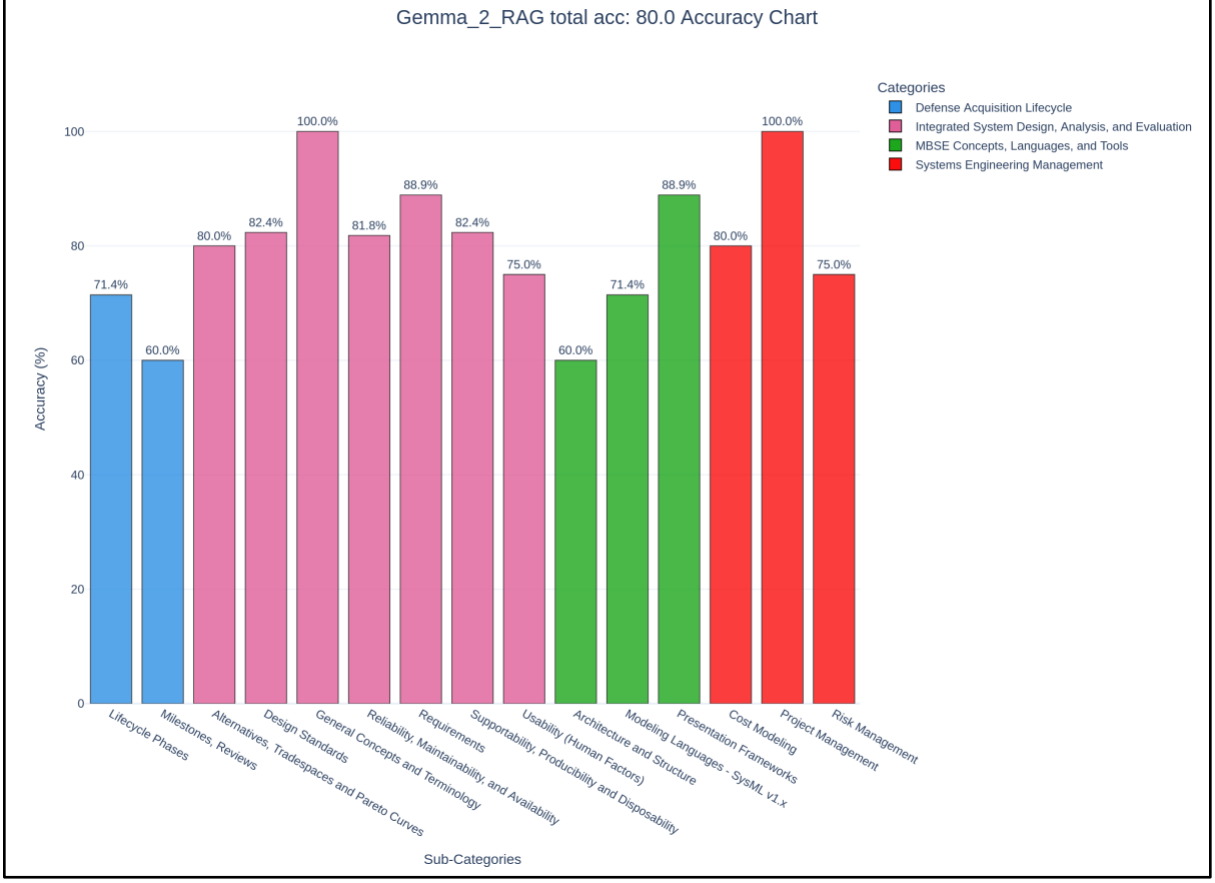
With the custom corpus we were able to reduce error rates by 28.61% for Llama 2 and by 38.28% for Orca and 23.81% for Gemma 2. This drastic improvement shows that given relevant and concise information these models can better answer technical field specific question in System engineering, some examples of text chunks from this corpus are:

- “Concept of Uncertainty in Systems Engineering: Uncertainty in systems engineering often refers to unpredictable outcomes due to incomplete information or inherent variability, which can impact system behavior and decision-making.”
- “Decision Management in SEM: Decision management in SEM incorporates risk, opportunity, and stakeholder input, ensuring well-rounded, informed choices.”
- “Logical Architecture in System Definition: Logical architecture outlines functional relationships within a system, providing a structured, non-physical view of its components.”
- “Cost Drivers in COCOMO Model: In COCOMO, product complexity is a significant cost driver, influencing resource allocation and project planning. “


Mistral_7B_RAG total acc: 85.0 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
0
20
40
60
80
100
71.4%
60.0%
80.0%
58.8%
100.0%
90.9%
100.0%
94.1%
83.3%
100.0%
95.2%
88.9%
80.0%
100.0%
62.5%
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management
Sub-Categories


Orca_2_RAG total acc: 86.88 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
0
20
40
60
80
100
92.9%
40.0%
80.0%
70.6%
100.0%
90.9%
94.4%
82.4%
100.0%
100.0%
90.5%
88.9%
80.0%
100.0%
75.0%
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management
Sub-Categories


Llama_2_13B_RAG total acc: 84.38 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
0
20
40
60
80
100
71.4%
60.0%
80.0%
64.7%
100.0%
81.8%
100.0%
94.1%
91.7%
80.0%
90.5%
77.8%
80.0%
100.0%
75.0%
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management
Sub-Categories


Gemma_2_RAG total acc: 80.0 Accuracy Chart
Categories
Defense Acquisition Lifecycle
Integrated System Design, Analysis, and Evaluation
MBSE Concepts, Languages, and Tools
Systems Engineering Management
Accuracy (%)
0
20
40
60
80
100
71.4%
60.0%
80.0%
82.4%
100.0%
81.8%
88.9%
82.4%
75.0%
60.0%
71.4%
88.9%
80.0%
100.0%
75.0%
Lifecycle Phases
Milestones, Reviews
Alternatives, Tradespaces and Pareto Curves
Design Standards
General Concepts and Terminology
Reliability, Maintainability, and Availability
Requirements
Supportability, Producibility and Disposability
Usability (Human Factors)
Architecture and Structure
Modeling Languages - SysML v1.x
Presentation Frameworks
Cost Modeling
Project Management
Risk Management
Sub-Categories

Custom GPT-4 corpus

Concept of Uncertainty in Systems Engineering: Uncertainty in systems engineering often refers to unpredictable outcomes due to incomplete information or inherent variability, which can impact system behavior and decision-making. Risk Identification in Systems Engineering: Effective risk identification combines historical insights and real-time data, often using brainstorming with experts, trade studies, and analysis of project-specific details. Risk Handling in Systems Engineering: Managing risks involves choosing suitable strategies (accept, avoid, transfer, or mitigate), evaluating impacts, and ensuring selected approaches align with project goals. Probability Distribution for Random Events: Certain probability distributions, like the exponential distribution, are used to model random events occurring without memory, such as time between failures. Redundancy Mechanisms in Reliability Block Diagrams: Reliability Block Diagrams may represent different redundancy types, including active and standby redundancies, used to enhance system reliability. FMECA and Fault Tree Analysis: FMECA is a tool used early in the fault analysis process to identify potential failure modes, which then informs fault tree structures for detailed risk analysis. Initiating Event in Event Tree Analysis: Identifying the initiating event involves questions about the nature and consequences of the event, helping to determine its root cause and possible outcomes. Unexplained Failure in Systems Engineering: Unexplained failures are anomalies with unknown causes, often challenging to replicate, particularly in high-reliability systems. Top-Down Reliability Analysis: Methods like FMEA start at the system level and work down to components, mapping out how failures affect each level. Core Aspects of Systems Engineering: Systems engineering focuses on managing complex, multidisciplinary systems with constraints on resources, ensuring these systems meet design and operational requirements. Visualizing Causal Relationships in Systems: Tools like causal loop diagrams are used to map interdependencies and feedback loops, aiding in understanding complex, systemic behavior over time. Model Verification vs. Validation: In systems engineering, verification ensures a model is built correctly to specifications, while validation confirms it meets the intended purpose or requirements. Functional vs. Non-functional Requirements: Functional requirements describe what a system must do, while non-functional requirements focus on quality attributes like performance, reliability, and usability. 'Time-bound' in SMART Requirements: In SMART criteria, 'Time-bound' specifies the deadline or schedule by which a requirement or goal should be met, ensuring timely completion. Feedback in IDEF0 Models: IDEF0 models use feedback arrows to represent information flow, often from bottom right to top left, showing corrective actions or information needed for process improvements. Project Metrics (MOEs) in Systems Engineering: MOEs define high-level requirements and are essential for evaluating alternative solutions based on operational needs and system objectives. Generic vs. Instantiated Physical Architecture: A generic physical architecture offers a blueprint, while an instantiated one specifies actual components, tailoring designs to specific

systems. Value Curve Shapes in MAVA: In Multi-Attribute Value Analysis, curve shapes illustrate how value shifts from minimum to maximum levels, reflecting stakeholder preferences. Purpose of Validation in Systems Engineering: Validation aims to confirm that systems meet user needs and performance criteria, focusing on functionality and user satisfaction. Supportability in Systems Engineering: This refers to how logistics and maintenance can keep the system operational, reducing downtime and ensuring long-term use. Domains of Systems Approaches in SEBoK: Systems approaches apply to various fields, including technology, civil infrastructure, and organizational systems, fostering a holistic view. Systems Science in Engineering: Systems science leverages universal principles to tackle complex, interdisciplinary challenges, often spanning both physical and social domains. Systems Thinking in Problem-Solving: This approach views problems holistically, emphasizing interconnected elements and the behavior of systems over time. Life Cycle Processes in Systems Engineering: Life cycle processes guide systems from conception to retirement, ensuring they meet stakeholder needs throughout their operational life. Systems Engineering Management (SEM) vs. Project Management: SEM emphasizes technical aspects of system design, while project management covers broader planning, scheduling, and resource allocation. Decision Management in SEM: Decision management in SEM incorporates risk, opportunity, and stakeholder input, ensuring well-rounded, informed choices. Logical Architecture in System Definition: Logical architecture outlines functional relationships within a system, providing a structured, non-physical view of its components. Service Life Extension for Sustainability: Practices that extend operational life reduce waste, lower costs, and conserve resources, enhancing system sustainability. Systems Engineering for SoS: Systems of Systems (SoS) engineering focuses on leveraging existing systems' capabilities, maximizing functionality and inter-system integration. Technical Leadership in Systems Engineering Teams: Effective technical leadership involves adaptability, fostering innovation, and encouraging team collaboration. Usage of BDDs in Systems Engineering: Block Definition Diagrams (BDDs) help clarify structural aspects of a system, often used in architectural and performance analyses. Impact of Generalizations in SysML: Generalizations in SysML streamline design by factoring out shared features, making system models more efficient and reusable. Blocks and Internal Block Diagrams in SysML: Blocks define types for elements on IBDs, which focus on system interactions and component relationships. Connector Information in IBDs: In SysML IBDs, connectors can have names and types to specify the nature of connections, aiding in system definition. Notation for Use Cases in SysML: Use cases in SysML are typically represented by an ellipse, visualizing system functionalities from a user perspective. Include Relationships in SysML: In SysML, an include relationship is shown with a dashed line and open arrowhead, indicating an included function or use case. Actor Interaction in Use Cases: SysML represents actor and use case interactions through associations, indicating roles and participations. Role of Object Nodes in Activity Diagrams: Object nodes act as storage points, retaining state within a SysML activity

diagram, beyond just token flow. Purpose of Object Flows in SysML: Object flows indicate the flow of matter, energy, or data, visualizing process transitions in an activity. Sequence Diagram Frames in SysML: Sequence diagrams in SysML include frames like interactions, which illustrate specific system behaviors. Start and End in Sequence Diagrams: Execution specifications mark the beginning and end of actions within SysML sequence diagrams, showing behavior flow. State Machine Diagram Timing: State machine diagrams can be developed at various lifecycle stages to model system states, allowing iterative refinement. Transition Constraints in State Machines: Transitions in SysML state machines generally cannot cross different regions, focusing on internal system dynamics. Constraint Parameters in SysML: Constraint parameters allow SysML models to incorporate specific variables, defining operational limits or dependencies. Notation for Binding Connectors: SysML binding connectors use a solid line without arrowheads to indicate equality constraints between elements. Crosshair Notation in Package Diagrams: In SysML, crosshair notation shows dependencies between packages, highlighting cross-package relationships. Purpose of View Packages: View packages focus on specific stakeholder perspectives, encapsulating relevant elements and diagrams. Purpose of Trace Relationships: Trace relationships in SysML indicate a requirement's origin or justification, connecting it to its purpose. Impact of Rationale Documentation: Rationale documentation provides context, enabling informed decisions by clarifying design choices and historical considerations. Capturing Rationales in SysML: Rationales can be annotated in SysML using a dedicated stereotype, offering insight into design decisions. Life-cycle Definitions and Communication: Detailed life-cycle phase definitions enhance documentation, aiding in clear communication of system progress. Cost Drivers in COCOMO Model: In COCOMO, product complexity is a significant cost driver, influencing resource allocation and project planning. Advancemanship in Software Projects: Techniques like iterative prototyping aid in development, enhancing project adaptability and reducing future risk. Impact of Anticipatory Documentation: Early documentation mitigates logistical and support issues, potentially lowering long-term project costs. Presentation of Unquantifiable Criteria: Effective presentation of qualitative factors aids decision-making, providing a visual basis for non-measurable criteria. Timing of System Requirement Review (SRR): SRR is conducted to establish foundational requirements early, aligning system objectives with stakeholder needs. Interoperability Verification: During technical review, interoperability assessments ensure system compatibility with existing infrastructures. Differences in Technical Audits: A system audit focuses on component integration, while an FCA targets individual function verifications. Role of Capability Development Document (CDD): CDD outlines key capabilities, typically approved to solidify a system's functional direction. Integrated Product Support (IPS) in Sustainment: IPS elements facilitate system support, maintaining operational readiness and addressing sustainment needs. Significance of Critical Design Review (CDR): CDR evaluates whether a design meets specifications, ensuring readiness for the development and testing of a

prototype system. Addressing System Obsolescence in Post-Production: This phase includes managing documentation and updates to ensure system relevance and compatibility as technology advances. Purpose of Integrated Baseline Review: This review assesses whether project baselines align with program objectives, checking cost, schedule, and performance goals. Impact of Anti-Tamper Technologies: Anti-tamper features are integrated to protect system integrity, especially in defense systems, enhancing security over the lifecycle. Timing of Configuration Management in Lifecycle: Configuration management starts early to control design changes, ensuring traceability and accountability throughout development. Role of Supply Chain Management in Risk: Effective supply chain management minimizes risks by monitoring suppliers and performance to maintain reliability and compliance. Safety and Occupational Health in Acquisition: Integrating safety and health standards during design ensures compliance and reduces operational risks for defense systems. HSI Domain of Survivability: Human System Integration in survivability addresses health risks, maintaining system readiness and user safety in deployment. Total System Approach in HSI: This approach improves supportability by designing with human needs in mind, enhancing system maintenance and usability. Noise Level Management in Habitability: Habitability design considers noise reduction to maintain crew performance and create a comfortable environment. Program Strategies and RFP Relationship: Program strategies inform RFP requirements, aligning project needs with vendor capabilities and ensuring relevant bids. Program Strategy Approval Process: This process involves evaluating strategic considerations to align project goals with technical specifications and budgeting.

No relevancy Mistral 7B results

Although the impact is less pronounced compared to the relevance results, the RAG implementation still improves Mistral's performance, increasing its accuracy from 79.38 to 86.88. This corresponds to a 36.37% reduction in the error rate, highlighting the benefit of incorporating RAG into the system.