Naïve RAG system evaluation with RAGAS

Sammy Cayo

Executive Summary:

After evaluating twelve RAG systems using the RAGAS framework to reduce hallucination for generated responses, the findings suggest chunking strategies significantly improve retrieval and response quality for marketing and research focus domains. Using a mean weighted evaluation score based on semantic similarity, Rouge-L, and BERT Score, the results show advanced chunking techniques can enhance system performance.

Introduction:

The primary objective of this report is to implement Retrieval Augmented Generation (RAG) systems using LangChain to address the seven failure points of RAG systems (Barnett, 2024). To achieve this, I conducted a comprehensive evaluation using the RAGAS framework, with the goal of reducing hallucination in generated responses. This analysis includes an in-depth evaluation of adjustments to core components of the indexing and querying process of twelve RAG systems. The adjusted components include different embedding models, prompt templates, large language models, chunking strategies, and additional pipelines for efficacy and safety. While some of these additional pipelines showcase promising results, due to compute constraints, they were not added to the final twelve RAG systems for testing.

The dataset used for testing and evaluation consists of 75 question/answer pairs, with each question having a set of ground truth reference answers for marketing and research. A sample table of questions and gold answers is in Appendix F.

Lastly, this report can be used as a proof of concept for organizations looking to improve their search and question answering capabilities for diverse audiences. The primary objective of this evaluation is to enhance communication between stakeholders, including engineering and marketing departments.

Key Findings:

Key findings of this evaluation include:

- Chunking strategies have the biggest influence of system performance with two of the best RAG models of each group using Unstructured.io's advanced, 'by_title' chunking algorithms.

- Prompt template construction plays a critical role in generating desired output from large language models that can be useful for communicating with different audiences.
- It's important to use semantic level metrics for evaluating LLM responses.
  - Traditional metric like Rouge Score failed to capture deeper level understanding between the generated answers and the reference text.
- Using a mean weighted evaluation score of three metrics in our RAGAS framework – Semantic Similarity, Rouge-L, and BERT Score – our research systems' results show greater consistency.

Methodology:

Technical Approach

RAG systems consist of two major processes: the index process and the query process. Within these two processes, there are several points of failure that can occurs when using a RAG system. To address these failure points, I tested two separate embedding models: multi-qa-mpnet-base-dot-v1 and all-mpnet-base-v2, which were used for storing and retrieving document chunks. The baseline RAG system – the multi-qa-mpnet-base-dot-v1 embedding model – was seeded with documents from the three data sources: ArXiv, Wikipedia, and Lillianweng blog articles. To chunk and store these documents, the baseline systems used LangChain's recursive character text splitter method with a chunk size of 128 characters and 0 overlap.

There were two separate baseline systems with different large language models: Mistral-7B-Instruct-v0.2, which loaded directly into memory hosted on Google Colab, and Cohere command model, accessed through LangChain's ChatCohere module with an API key. The baseline retriever embedding return k = 4 documents in each invocation of a query using cosine similarity search.

Critical to our objective of generating responses with different audience, I created prompt templates with two user personalities, one for marketing and one for research. Both prompts were pass through mistral and rewritten with instructions for each audience, the details can be found in the table in the Appendix A. This process resulted in a clearly defined prompt that effectively aligns with both research and marketing objectives.

Subsequently, there were several additional pipelines experimented with that failed to make the final test systems; these were: query re-writer, doc re-ranker, embedding filter, and relevancy filter. The query re-writer was introduced as a safety against malicious intent and instructed to remove harmful elements by rewriting the query. However, at runtime, this made the response generation incredibly slow and allowed the rewritten query to circumvent the prompt templates, resulting in similarities in both marketing and research responses.

Next, a LLM-based re-ranker was also introduced using LangChain LLMListwiseRerank module for ranking documents using a similarity threshold of 0.75. This re-ranker required an additional API call to the Mistral 7b model and did not allow usage of the local LLM loaded in memory. Because of the cost constraints, this pipeline was not included in the test systems.

Furthermore, both an embedding filter and relevancy filter were set up to filter out chunks irrelevant to the query. This was meant to address the incorrect specificity failure by reducing the noise of the retrieved context for the LLM to produce higher quality responses.

I tested several additional pipelines to improve response generation. LangChain's string output parser and a self-defined output formatter function to parse out only the answers for each question, producing clean text output compatible with our RAGAS evaluation.

For our alternative system, Unstructured.io's "by_title" chunking strategy was used to chunk documents. This strategy groups chunks together by sections and preserves section boundaries, resulting in distinct chunks not shared between sections. This is a more advanced chunking method than the recursive text splitter and provides a smarter retrieval. For our document sources, this is an optimal strategy since our ArXiv, Wikipedia, and blog documents provide clearly defined headings and subheadings. Like our baseline multi-qa system and all-mpnet system, four unstructured RAG system was tested, two for marketing using both Mistral and Cohere as LLMs, and two for research. The full list of model configurations can be seen Appendix C.

Testing and evaluation:

Four metrics—Semantic similarity, Rouge Score, BERT Score, and a weighted mean—were used to evaluate the twelve RAG System. The weighted mean was computed using .40 * Semantic Similarity Score + .20 * Rouge Score + .40 * BERT Score. These metrics provide a well-rounded evaluation for scoring generated responses against the ground truth. Semantic similarity measures alignment of generated answers, using the embedding model as the evaluator embedding. Rouge Score, computed using rogue-L, finds the longest matching subsequence of words between the reference text and the response. It's a traditional n-gram example for evaluating machine-generated text. BERT Score used RoBERTa-large to evaluate the quality of text-generated answers using a similarity score between the response answers and the reference ground truth. In general, this determines how well the generated answers capture the semantics of the reference text.

For the Ragas framework, the evaluator LLM was a 1.58-bit quantized Llama 3.1 8B model. This was a local option after having issues with API service, causing the RAGAS evaluation to produce inconsistent results. The test results can be viewed in the plots in Appendix D for the twelve-model evaluation, and Appendix E for the best three models for both the research and marketing systems.

Results and Findings:

The test results presented in Appendix E and D demonstrate the superiority of the unstructured chunking system over the baseline multi-qa and all-mpnet configurations in both marketing and research. This outcome clearly indicates that chunking and retrieval play a pivotal role in the performance of RAG systems.

Lessons Learned:

Through this evaluation of RAG systems, the best way to improve performance is by implementing advance chunking strategies. In our finding, unstructured 'by_title' chunking strategies outperform other chunking specific methods regardless of LLM or domain focus. Using a mean weighted evaluation score based on semantic similarity, Rouge-L, and BERT Score, Unstructured RAG systems were the highest scoring systems for both marketing and research.

Challenges and Limitations:

The cost of testing these models can make it difficult to use this system at scale. It important to deeply think about the use case and if an alternative, pay-per-use service would be more cost effective.

Next Steps

With more time, it would be beneficial to experiment with various hyperparameters in the LLM to enhance its performance for each domain. Additionally, exploring the incorporation advanced components to enhance safety and efficacy could provide promising results.

Summary & Recommendations

This study examined the effectiveness of twelve RAG systems through the RAGAS framework, focusing on reducing hallucinations in generated responses. The research demonstrated that the advanced chunking method plays a crucial role in enhancing retrieval accuracy and response quality for both marketing and research domains. This system has some risks, particularly that it lacks safety measures for malicious queries, prompt injections, and other harmful requests. In order to be used in production, it's important to provide guardrails against these potential risks. The next steps are to test more advanced components to improve the seven failure points of RAG systems for better performance.

References

Barnett, Scott, et al. "Seven failure points when engineering a retrieval augmented generation system." *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*, 14 Apr. 2024, pp. 194–199, https://doi.org/10.1145/3644815.3644945.

Apple Inc. (2024). *Apple Intelligence: Foundation language models*. Apple Machine Learning Research. Retrieved December 10, 2024, from https://machinelearning.apple.com/research/apple-intelligence-foundation-language-models

Google Research. (2024). *Gemini: The next-generation AI*. Retrieved December 10, 2024, from https://ai.google/tools/gemini

Perplexity AI. (2024). *Perplexity AI documentation and updates*. Retrieved December 10, 2024, from https://www.perplexity.ai

Apple intelligence was used to proofread the final report, checking for grammatical errors. Perplexity was used to debug code in the assignment notebook, and Gemini was used to help create plots used in this report.

Appendix

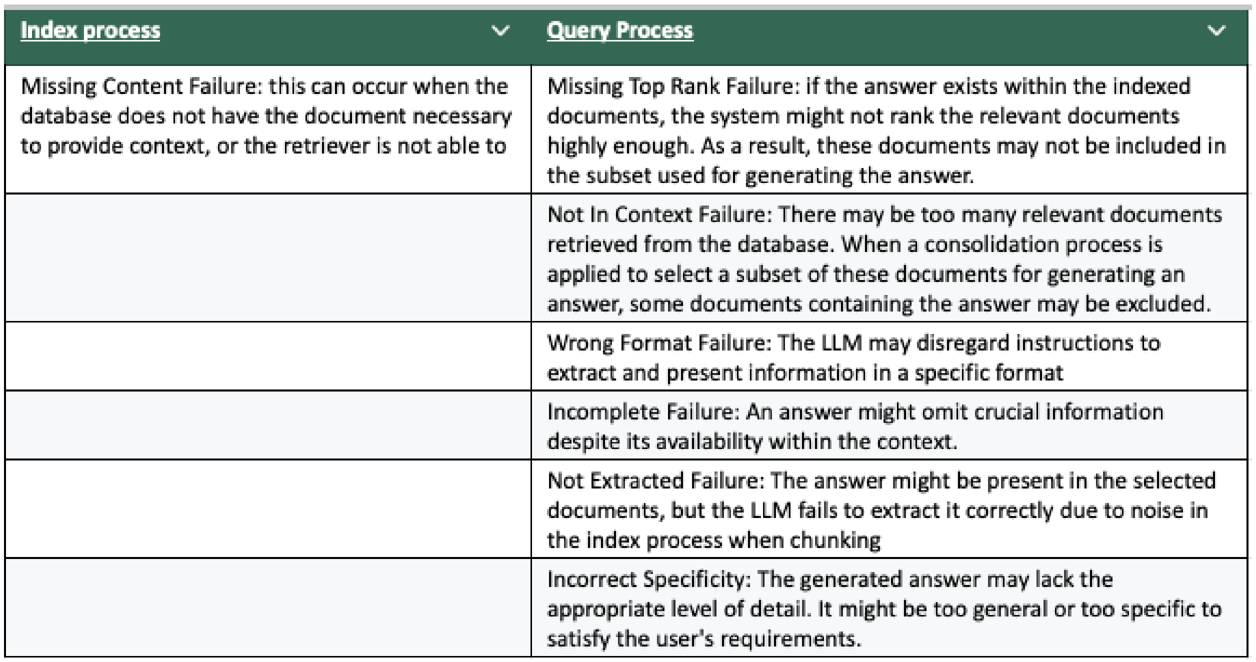
Appendix A: Prompt Templates

Instruction	“Context: {context} Use the context to answer the question in a neutral way. If you are not certain about the answer to the question, respond with ‘I need more context to answer the question’, or something indicating you are not able to answer the question base on the retrieved context. Do not start the answer with ‘based on the following context’, use natural language to answer the question. Question: {question}”
Draft marketing prompt	“You are a marketing executive with expertise in digital strategies, branding, and product positioning, providing responses to Generative AI-related queries. You should tailor your responses to the marketing department, and only include high-level technical details of AI architectures. Ensure the responses are informative and factual. You are doing a great job, I believe in you.”
Draft research prompt	“You are a highly skilled engineer with expertise in Generative AI, system architecture, and technical problem-solving. Provide detailed, accurate, and context-specific responses tailored to be clear and digestible for executives. You are allowed to include technical jargon to explain any concepts in the question and to provide a low-level explanation. Do not start the answer with "based on the following context", use natural language to answer the question. You are doing a great job, I believe you can do this”

Marketing Rewrite Instructions	"[INST] I want you to rewrite and improve this marketing prompt template: {draft_marketing_template} for a RAG system. The goal is to improve the responses of the RAG for the internal marketing department when given a set of questions about generative AI. the audience for these responses are executives of a tech company who will make decision based on the responses. start your new response with '[/INST]' """
Research Rewrite Instructions	"[INST] I want you to rewrite and improve this research prompt template: {draft_research_template} for a RAG system. The goal is to improve the responses of the RAG for the internal engineering department when given a set of questions about generative AI. the audience for these responses are executives of a tech company who will make decision based on the responses. make sure to start your new response with '[/INST]' "

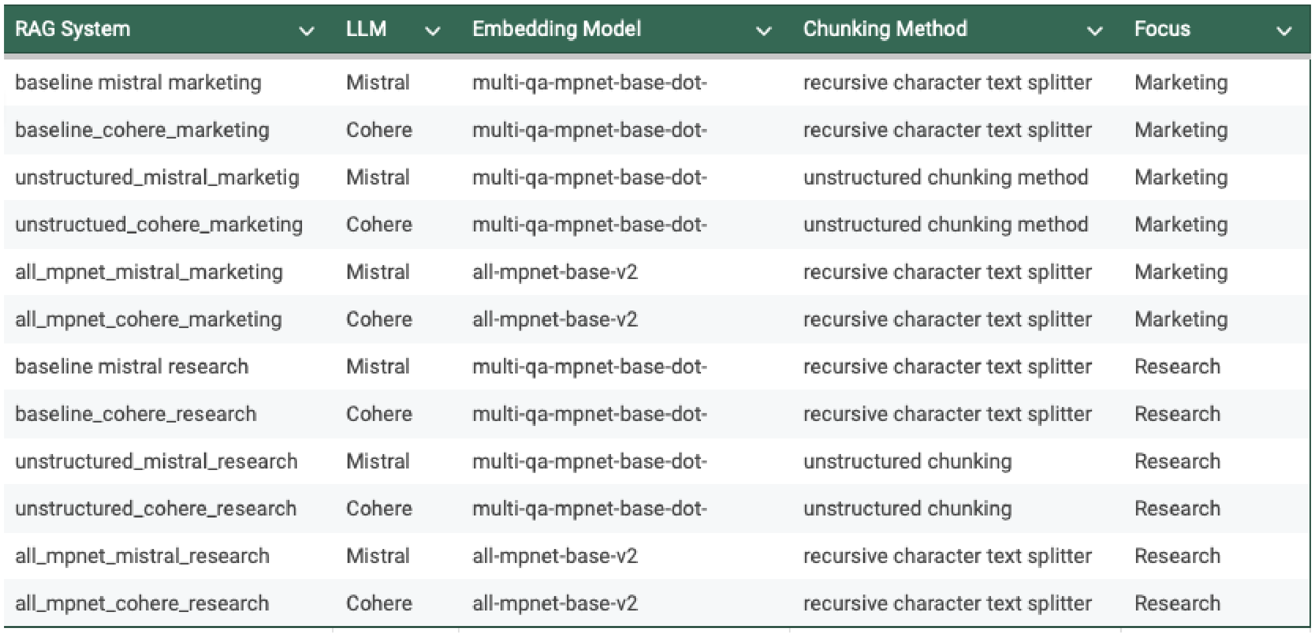
Rewritten Marketing prompt	"As a marketing expert specializing in digital strategies, branding, and "'product positioning, I am here to deliver accurate, high-level information "on generative AI for our esteemed marketing department. My answers will be "geared towards strategic decision-makers within your organization, focusing "on the business implications and benefits of integrating generative AI into "marketing efforts. Rest assured that all responses provided adhere to the "highest standards of factuality and professionalism. Your trust in my "abilities is well placed.
Rewritten Research Prompt	"As a highly competent engineer specializing in Generative AI, system design, and intricate issue resolution, my aim is to deliver precise, comprehensive, and executive-friendly answers regarding the application of generative AI within the context of an R AG (Rapid Application Development) system. I will utilize appropriate technical terminology where necessary to clarify concepts, while maintaining a conversational style to ensure clarity for non-technical audiences. My objective is to enhance the accuracy and accessibility of the R AG's responses when addressing queries related to generative AI from the internal engineering department or other executives in our technology organization.

Appendix B: Seven failure points



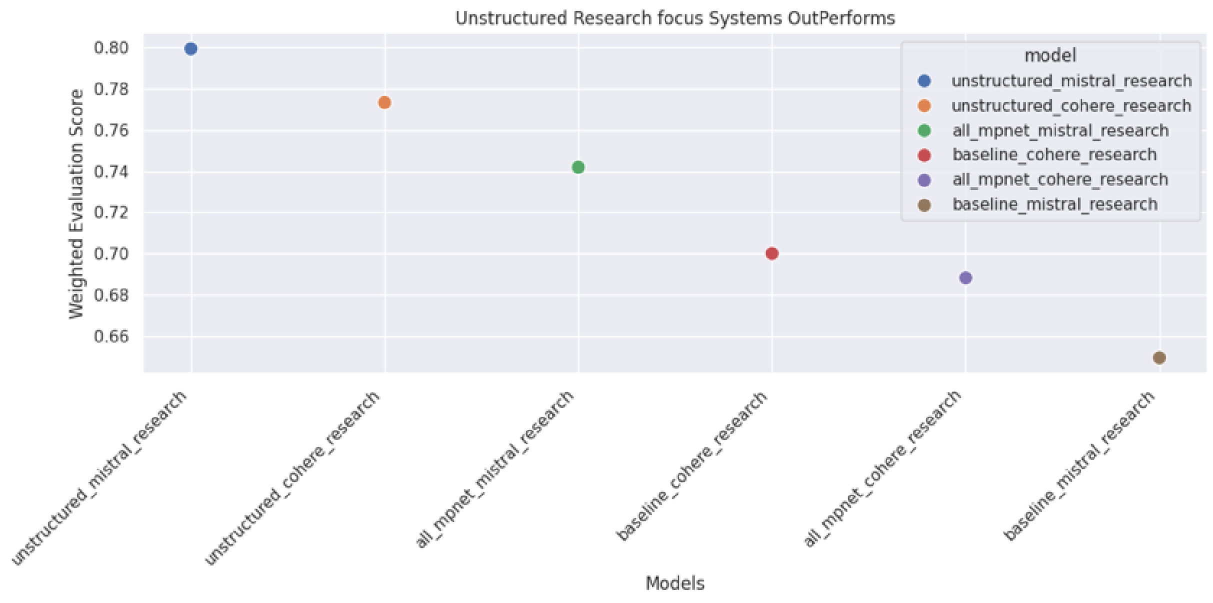
Index process	Query Process
Missing Content Failure: this can occur when the database does not have the document necessary to provide context, or the retriever is not able to	Missing Top Rank Failure: if the answer exists within the indexed documents, the system might not rank the relevant documents highly enough. As a result, these documents may not be included in the subset used for generating the answer.
	Not In Context Failure: There may be too many relevant documents retrieved from the database. When a consolidation process is applied to select a subset of these documents for generating an answer, some documents containing the answer may be excluded.
	Wrong Format Failure: The LLM may disregard instructions to extract and present information in a specific format
	Incomplete Failure: An answer might omit crucial information despite its availability within the context.
	Not Extracted Failure: The answer might be present in the selected documents, but the LLM fails to extract it correctly due to noise in the index process when chunking
	Incorrect Specificity: The generated answer may lack the appropriate level of detail. It might be too general or too specific to satisfy the user's requirements.

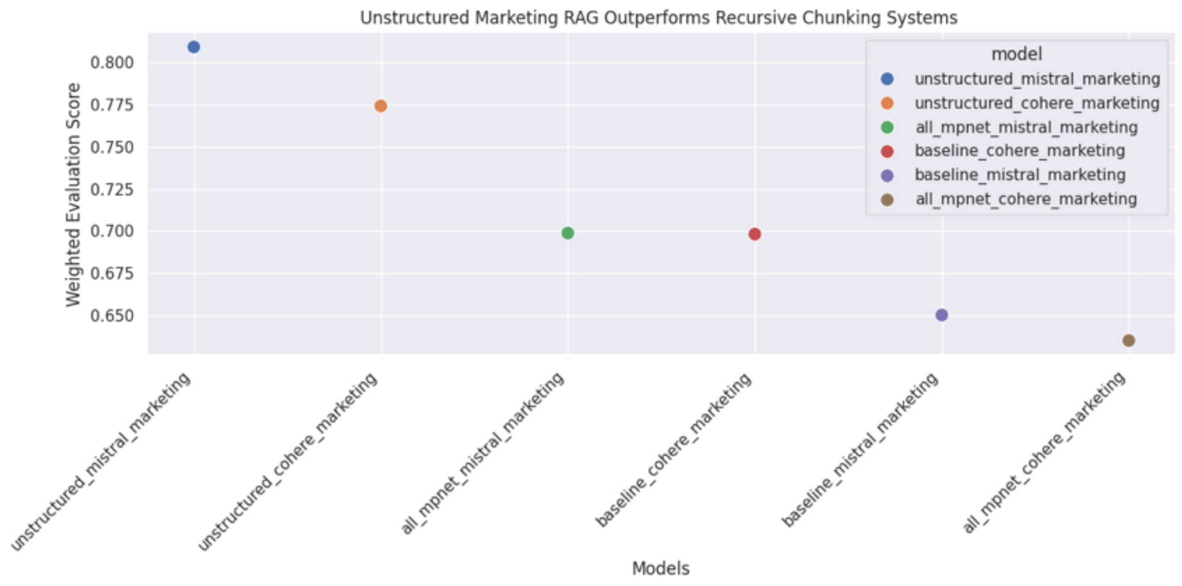
Appendix C: RAG System table



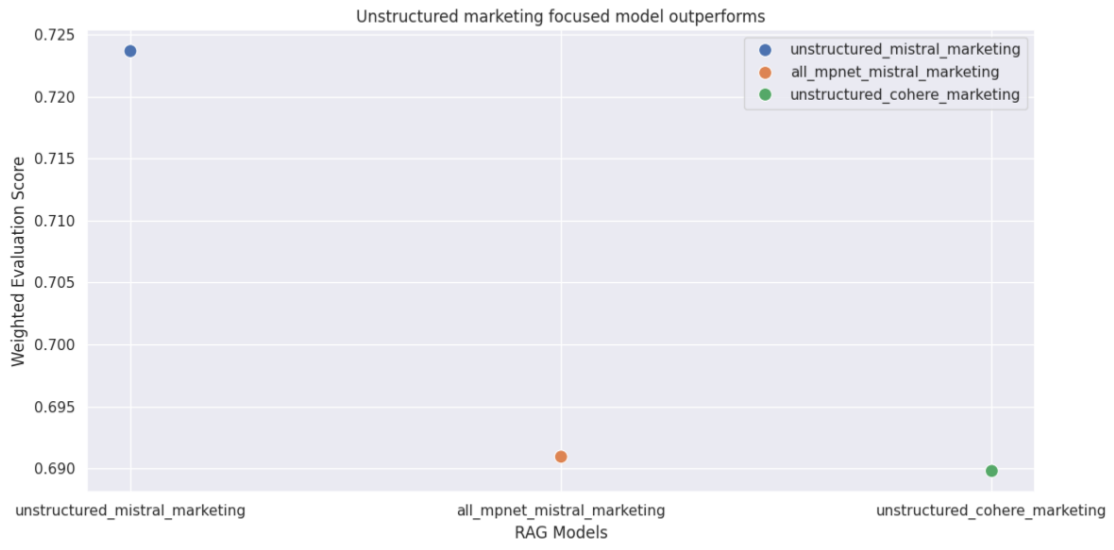
RAG System	LLM	Embedding Model	Chunking Method	Focus
baseline mistral marketing	Mistral	multi-qa-mpnet-base-dot-	recursive character text splitter	Marketing
baseline_cohere_marketing	Cohere	multi-qa-mpnet-base-dot-	recursive character text splitter	Marketing
unstructured_mistral_marketig	Mistral	multi-qa-mpnet-base-dot-	unstructured chunking method	Marketing
unstructued_cohere_marketing	Cohere	multi-qa-mpnet-base-dot-	unstructured chunking method	Marketing
all_mpnet_mistral_marketing	Mistral	all-mpnet-base-v2	recursive character text splitter	Marketing
all_mpnet_cohere_marketing	Cohere	all-mpnet-base-v2	recursive character text splitter	Marketing
baseline mistral research	Mistral	multi-qa-mpnet-base-dot-	recursive character text splitter	Research
baseline_cohere_research	Cohere	multi-qa-mpnet-base-dot-	recursive character text splitter	Research
unstructured_mistral_research	Mistral	multi-qa-mpnet-base-dot-	unstructured chunking	Research
unstructured_cohere_research	Cohere	multi-qa-mpnet-base-dot-	unstructured chunking	Research
all_mpnet_mistral_research	Mistral	all-mpnet-base-v2	recursive character text splitter	Research
all_mpnet_cohere_research	Cohere	all-mpnet-base-v2	recursive character text splitter	Research

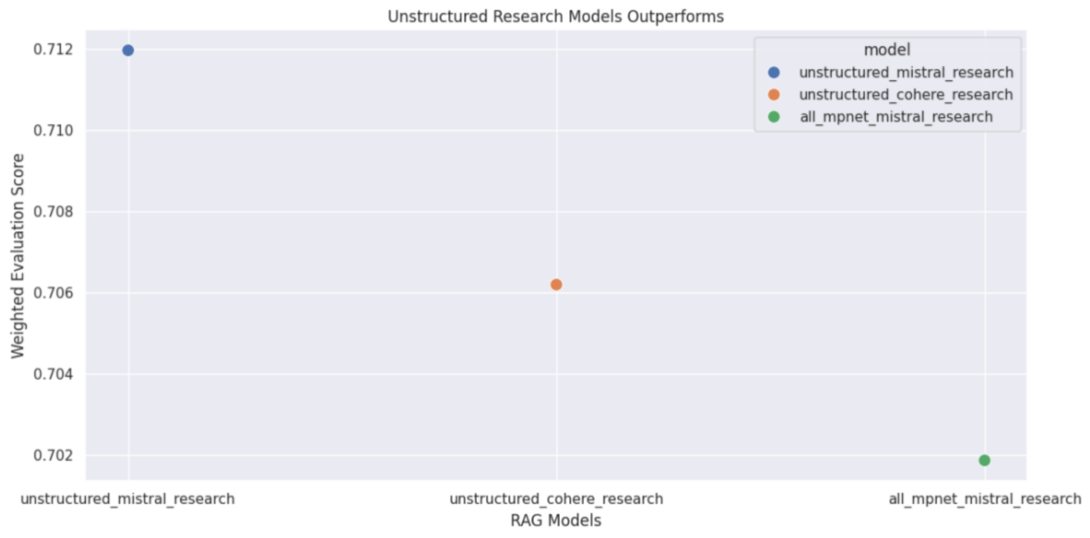
Appendix D: RAG System Evaluation



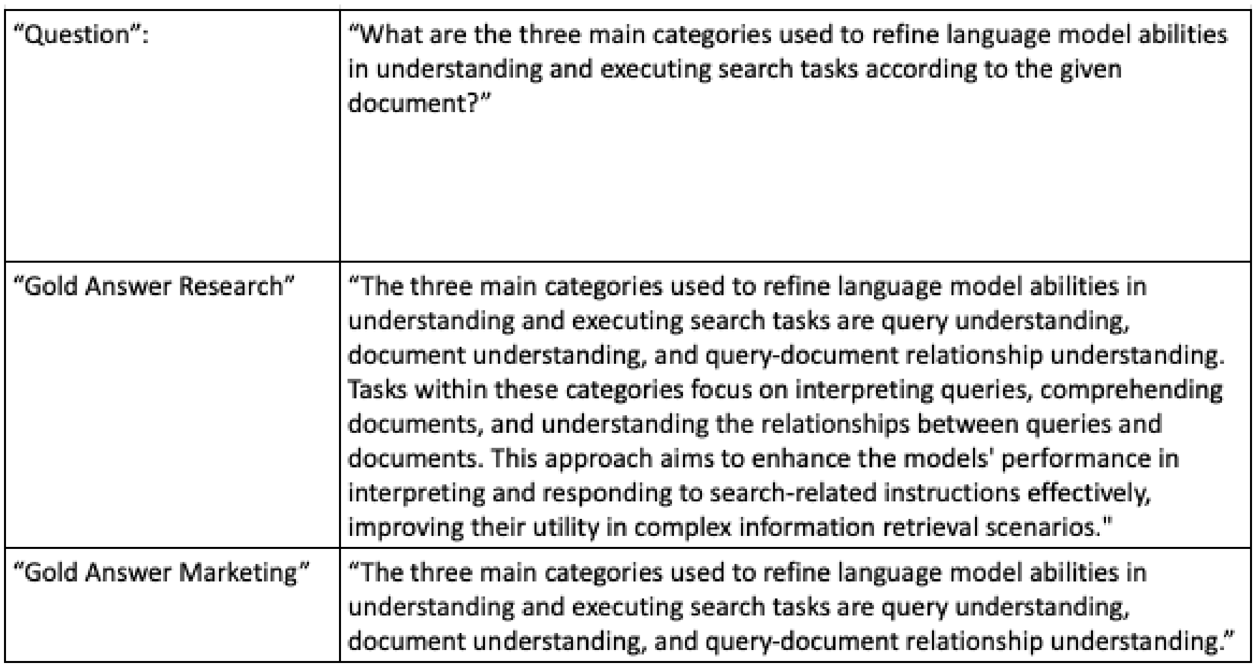


Appendix E: Best Model evaluation





Appendix F: Sample Question



“Question”:	“What are the three main categories used to refine language model abilities in understanding and executing search tasks according to the given document?”
“Gold Answer Research”	“The three main categories used to refine language model abilities in understanding and executing search tasks are query understanding, document understanding, and query-document relationship understanding. Tasks within these categories focus on interpreting queries, comprehending documents, and understanding the relationships between queries and documents. This approach aims to enhance the models' performance in interpreting and responding to search-related instructions effectively, improving their utility in complex information retrieval scenarios."
“Gold Answer Marketing”	“The three main categories used to refine language model abilities in understanding and executing search tasks are query understanding, document understanding, and query-document relationship understanding.”