The Conversational Network:

AI-powered Language Models for Smarter Cable Operations

A technical paper prepared for presentation at SCTE TechExpo24

Tyler Glenn
Principal Engineer
CableLabs
T.Glenn@CableLabs.com

Jason Rupe Ph.D.
Distinguished Technologist
CableLabs
J.Rupe@CableLabs.com

Kyle Haefner Ph.D.
Principal Architect
CableLabs
K.Haefner@CableLabs.com

Table of Contents

List of Figures

List of Tables

1. Introduction

Enabling technical talent in network operations has been a challenge since the first network was created. As technicians and engineers figure out how to plan, engineer, manage, and repair a communication technology, the next technology comes around and resets the learning curve. Like Sisyphus, it can feel like the rock rolled back down the hill and our task is to try again to roll the rock back up the hill for the next technology.

Frustration aside, the challenge is long standing, continuously evolving, and always becoming more challenging. As we get better at training and educating the workforce, and get better at managing and maintaining our networks, the network gets harder to manage and maintain as the performance bar is raised too. What is possible improves, so the bar floats above the performance line.

Training the workforce how to do the job is one part of the system. Another part is determining what is the right action to take. Knowing how to use a hammer is one part of training; when and where to use the hammer or not to use the hammer is another important part. The challenge is to train for situations that are highly variable and help them make good decisions.

But good decision making takes time to learn and be reinforced. Repetition is needed.

Operators can't afford to apprentice, meaning have an unexperienced person shadow an experienced person to learn. The usual approach to training is to instruct someone on the how, which will involve some aspects of the where and when, but expect they have learned over time (degrees, experience in related role, etc.) to get the rest of the way there. That's not always easy, possible, or the outcome.

Another approach is to create access to an expert. In the center, that can happen to a degree, but the expert may not always understand the situation and may not always have all the information needed to make the right decision.

Generative AI (GAI) presents a new approach: accumulate the knowledge of experts, encode it for fast access, incorporate situational information, and create the equivalent of an expert assistant to help the person do a better job. But instead of being simply a search engine, GAI provides the information in a way that is immediately useful to the human; instead of providing a likely answer or set of sources to read, it is provided as part of a conversation between the user and an expert.

This is a compelling promise, and as it turns out, it seems very reasonable to expect it will contribute well to this problem.

2. Background - In the Beginning There Were LLMs

As we are caught in the eye of the storm by this wave of rapid advancements in large language models (LLMs), the question arises: What use can we make of this new technology in the Cable Industry? The capabilities of LLMs seem endless, with their ability to generate text on the fly and seamless creation of extremely convincing output. One might think they bridge a gap in a much-needed area of AI, the ability to mimic human thought. However, as we examine the output of these LLMs closer, we find God is in the detail. While the LLM’s output is convincing, a deeper inspection of the contents often finds a multitude of problems. Traditional LLMs suffer from various problems including hallucinations, inaccuracies, poor reliability, hackability, lack of accountability and grounding evidence. This is often dangerous to the untrained eye, spreading misinformation and sometimes downright false information in an often well worded and extremely convincing explanation.

All of these problems complicate the task of retrieving factually correct information from LLMs and the question arises, “How do we counteract these problems?”. One option is to continually train the LLM on larger amounts of information. Research has shown that LLM accuracy directly correlates to input text data size. However, the cost of training LLMs on large datasets can enter the realm of millions of dollars. While models will continue to improve as more data is incorporated, they will continue to suffer the same problems. Fine tuning can reduce the costs and provide adequate results, but often suffer similar problems as base models. We look towards a quicker, less costly and effective method of producing accurate responses. Enter Retrieval Augment Generation or RAG.

What is RAG and how does it help our situation? The process of RAG is akin to an open-book test. As anyone knows, using a textbook on a test yields much better results as compared to a memory-based test. RAG replicates this process by providing the LLM with a set of relevant context chunks allowing the LLM to make more accurate completions as shown in Figure 1. In addition to providing the LLM relevant information relating to the user’s query, RAG can also be used to incorporate up-to-date information without the need to re-train.

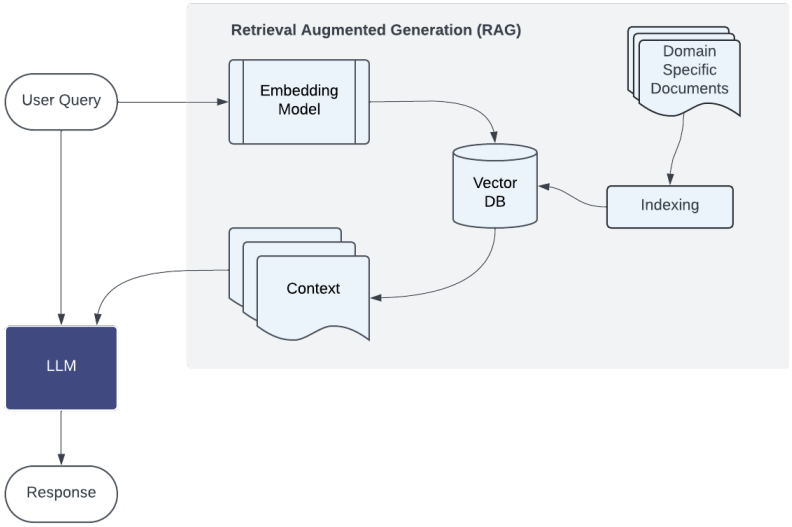

Figure 1 - Basic Example of Retrieval Agumented Generation

3. From RAGs to Riches

3.1. The Goal of Our Efforts

The goal was simple at the start: build a CableLabs Domain Expert LLM. On the surface it seems simple; but as we dug in further, we found the idealized notion of an LLM responding accurately to all manner of user questions about DOCSIS® networks and SCTE to be incredibly challenging. The main crux of the challenge was the extremely compelling answers outputted by chat LLMs, which often included false, inaccurate or blatantly made-up information on the subject. While to an advanced and knowledgeable user this was merely an inconvenience, to the unexperienced users lacking proper knowledge of the subject these answers are dangerous and without proper fact checking could lead the user down a rabbit hole of misinformation.

As we began testing existing LLMs including GPT 3.5, GPT 4, Llama2, Mixtral, Mistral and others we found they all suffered from these issues. During our initial research of the space, we came across a new technique of the time, RAG. At the time RAG promised better results by providing the LLM factually relevant information as context into the LLM. We began experimenting with RAG to see how it impacted the output by testing existing solutions. There were few solutions at the time, and most were in their infancy. In our initial manual testing and review of RAG frameworks we found the answers to be more factually grounded in the context material, while still containing errors. We noticed the quality of the context dramatically impacted the quality of the answer.

We started our initial testing by uploading DOCSIS specs into popular RAG frameworks of the time including ChatGPT [3], Danswer [4], H2O-GPT [5], Open WebUI [6], Anything LLM [7], BionicGPT [8], AutoGPT [9]. Danswer showed the most promising results and we used it for the initial testing. We started by compiling a list of questions about the DOCSIS 4.0 spec and asking questions to the LLM. While the RAG augmented LLM answers provided better results than non-RAG answers, through manual inspection we often found the answers missing information and partially incorrect.

3.2. Initial Experimentation and Findings

One of the first errors encountered provided an eye-opening revelation. In this question the LLM referred to a table, however the numbers in the LLM outputted response all had the number one appended to them. Perplexing at first, after looking into the root cause of the problem we found the original specification document had footnotes on those values in the table. As part of the process of RAG, the input documents must be indexed, a process involving converting the documents to text. The raw text is then chunked, run through an embedding model and input into the vector database. In the initial conversion of the specification, we found the footnotes were being converted into text improperly and were simply appended to the values in the raw text. This initial finding was an eye-opening experience – we realized the conversion process was of vital importance. As anyone might expect junk input yields junk output from the LLM.

As our research and experimentation solidified, we noticed the need to develop our own framework to better control the process, conduct more advanced experiments, and implement feature improvements. In the beginning we started with a simple web chat interface to a LangChain [10] backend implementing a simple version of RAG. We continued our testing on our framework and found a need for a set of test questions and expected answers to provide consistency and repeatability. With these questions we began uploading PDF versions of the published DOCSIS 4.0 and 3.1 specifications and running the test questions. We then had experts rate the answers. While the initial testing was not deeply rooted in the scientific process, we did make several important findings we were later able to verify with appropriate testing.

Another early discovery we found was in the representation of tables and figures. When asking questions about figures we found the LLM unable to answer correctly. After reviewing the PDFs we found most figures and some tables were lost in the text conversion process. This was due to the tables and figures being shown as images. Images, easy to understand by humans, provide a rich set of knowledge that is often intuitive to understand. The old phrase goes, "A picture is worth a thousand words" and this certainly holds true for LLMs, as long as they can read them.

3.3. How to Score the LLM Output

With a RAG solution in place, we needed to empirically prove the LLM's RAG augmented outputs were better. As with everything else we first turned toward looking at existing solutions. We found RAGAS [11] and TruLens [12] to be two open-source solutions available for response validation. After integrating

RAGAS and TruLens with our RAG application we found the results less than appealing. Often the scoring would score low if the answer was worded differently or if the answer included extra information or too little information. This made the results inaccurate and unusable.

The inability to score the LLM's output in an automated fashion led to our first big breakthrough. A custom scoring solution was devised in which the LLM's response was fed into a separate scoring LLM. The scoring LLM was asked to judge the original answer based on the question, context, expected answer and scoring guidelines. As LLMs are language based and it was found they struggle with numbers, we found it more effective to use a scoring system based on letter grades. The LLM would provide a grade from A to E as well as a rationale for the scoring. The grade was then converted to a number 0% through 100%.

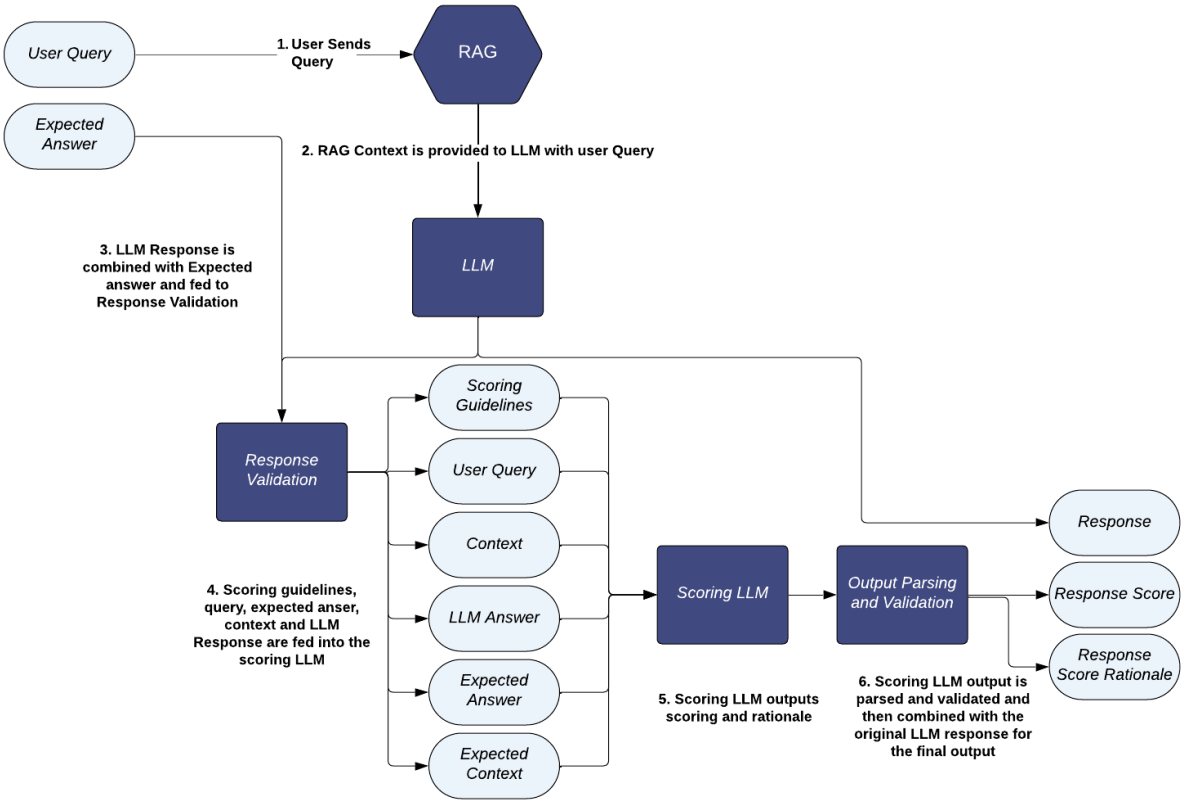

Figure 2 - Custom scoring architecture

Our custom scoring solution produced positive results and we immediately adopted it into use. However, as it was easy to tell our custom scoring solution was working we still needed to prove its accuracy. At this point we had grown our test questions to 119 questions ranging from easy to more difficult all relating to SCTE 280 [2]. To test the accuracy of our scoring we ran the questions through our RAG application to get responses and then had an expert judge the response on a scale of 0 to 10. In hindsight we should have had the expert score on the same scale of 0 to 5, however the scores were normalized after the fact. We then took the delta between the expert score and the automated score and averaged across all questions. We did this with a number of LLM models to generate a larger dataset. The results are shown in Table 1 – Scoring accuracy of Custom Scoring, RAGAS and TruLens as run on five test datasets with questions run on GPT 3.5, GPT 4, GPT 4o, Claude 3 – Sonnet, and Llama 3 and show our custom scoring solution to outperform RAGAS and TruLens getting an average of 92% accuracy.

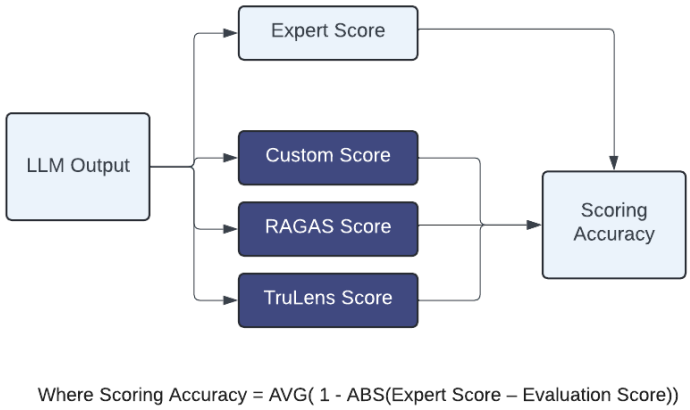

Figure 3 - Process of determining scoring accuracy

3.4. Determining the Best Document Format

As we gained more experience with error cases a trend started to emerge. We found tables, figures, formatting, and alt-text to all significantly impact the quality of the input context and thus the quality of the outputted answer. We began to dive in deeper to determine which formats provided the most favorable results. For each we ran one off tests to see how well various formats performed with LLMs. Later we then devised a set of tests to measure the response score measured in answer correctness for each format. Over time trends emerged and we began to build a picture of the best practices to use when formatting documents for LLM ingestibility.

Tables turned out to be the easier of the two problems to tackle. We were able to find the original Microsoft Word versions of the PDFs, which often had the original tables. The question then arose, which format was best suited for LLMs? We devised a test to determine the best format. We chose several text-based formats for tables including CSV, JSON, YAML, Markdown and AsciiDoc and the results were surprising. We started with a simple table from SCTE 280 and simple questions, on this test the responses were all correct, however in the second test we used a more complex table showing basketball statistics with more complex questions. Table shows the results of the complex basketball statistics table in which we asked a set of 20 questions about the table. For each format of the table we validated the RAG/LLMs answer to each question. While all formats were text based not all performed the same when fed to the RAG/LLM and we found AsciiDoc, Markdown and CSV to perform the best.

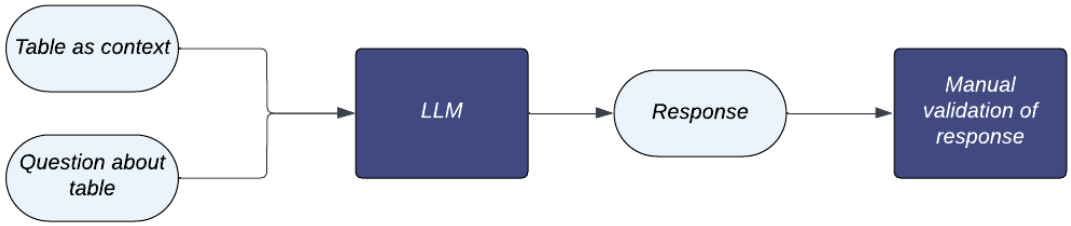

Figure 4 - Table format test process

Figures turned out to be a more challenging endeavor and encompassed everything ranging from flow diagrams to spectrum captures. We first started down the path of flow diagrams, the thought being to try testing a number of text-based formats. We found Mermaid and PlantUML to yield accurate results when fed to the RAG application. However, the process was currently a manual one, we needed to develop an automated method for the many numbers of documents we planned to convert in the future. We are currently running tests on the viability of figure formats, which we hope to share in the future at the presentation.

Looking into the options we found several tools claiming to convert images to text. Most of the tools available relied on OCR, computer vision or Multimodal Models. As we began testing image conversion, we found errors arising in the conversion. In one instance with a converted state diagram the OCR had combined the text in two separate boxes which were at the same vertical position in the diagram. The arrows in the state diagram were also completely disregarded and their meaning lost in the automated conversion. In some cases, the entire meaning of the figure was lost, and the only output was a garble of text. With the current automated processes falling short we turned to a manual process of creating intent-based summaries for the majority of figures unless a pure text version such as Mermaid was available.

During this time, we determined it would be best to generate a "Golden Document" to set the standard for how to convert specifications and standards into LLM ingestible documents. We chose SCTE 280 as our "Golden Document" to convert due to its lack of dependencies on other documents, relatively short length and proportion of figures and tables. With our success in AsciiDoc we decided to use AsciiDoc as the source of the document. The goal here being to create a carefully curated document providing the best context to an LLM. All of our testing would be performance based and would rely on testing one feature at a time. We began by converting the documents to AsciiDoc. We found the PDF versions to yield an imperfect conversion and moved to the original Word document format, which provided a more complete conversion of the original text, formatting, tables, and figures.

Pandoc [13] was used to convert the Word documents after which we post processed them. The tables were natively converted to AsciiDoc format. We started the process of manually converting figures and images to text for those without a direct text representation. An expert in the subject was asked to write alt-text for each figure to capture the intent of the figure as well as any important information. After completing the alt-text for SCTE 280 we tested the accuracy of the LLM output. Table 3 shows the results of alt-text vs non-alt text and Figure 5 shows a diagram of the test process. Table 3 shows the results of alt-text vs non-alt text and Figure 5 shows a diagram of the test process. We were relieved to find alt-text did indeed substantially improve the LLMs response to our test questions with answer correctness being on average 35% better with alt-text.

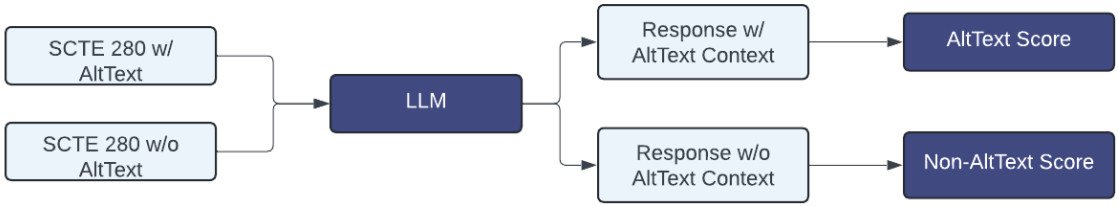

Figure 5 - Process for testing alt-text vs non alt-text answer correctness

After testing alt-text we turned toward removing formatting including page numbers, headers, footers and table of contents, all things that provide extraneous information and dilute the context quality provided to the LLM . During the retrieval process chunks of the document are returned with the highest likelihood of matching the user query. If these chunks have irrelevant, partial or misleading information in them it reduces the quality of the output from the LLM.

To prove the effectiveness of our curated "Golden Document" we ran our test questions on the SCTE 280 "Golden Document" vs SCTE 280 converted to text via LlamaParse [14]. The results are shown in Table 2. We found a 13% increase in answer correctness when using the "Golden Document". The reduction of formatting errors in the documents produced context chunks with more relevant information providing better responses to the questions. Removing formatting also reduced errors in the output answers due to oddities in the input context due to errors in formatting conversion.

3.5. Effectiveness of RAG vs Non-RAG

With the "Golden" SCTE 280 document and a validated scoring method we then set our sights to answering the final question – "Are RAG answers better than non-RAG answers?". We now had all the tools we needed to answer this hypothesis, a set of test questions, a "Golden" SCTE 280 document, and a validated automated scoring solution. Writing a test for this was simple – compare the answers with RAG enabled and RAG disabled using SCTE 280 as context.

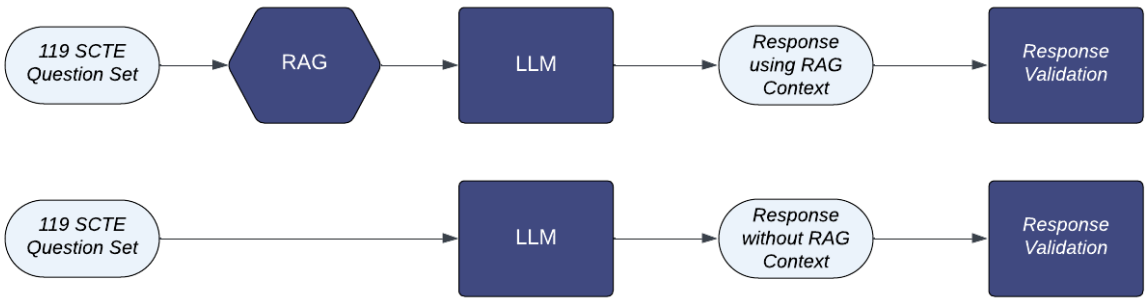

Figure 6 - RAG vs non-RAG test process

The findings were overwhelmingly positive, we found RAG did indeed significantly improve the LLM's answers. When run on five LLM's, GPT 3.5, GPT 4, GPT 4o, Llama 3 and Claude 3 - Sonnet using the

RAG application we found an average 16% improvement in answer correctness using RAG vs non-RAG. Refer to Table 5 for the results of the RAG vs non-RAG test.

4. Results

4.1. Scoring

For the custom scoring test, the 119 SCTE 280 document test question set was used and ran them through the RAG application for each of the of the models below. The answers for each model were then fed to each of the scoring methods. The answers from the model were also manually scored by an expert. To get the scoring accuracy the absolute value of the delta between the expert score and the automated score was taken and then subtracted from one to get a percentage. Refer to Figure 3 for a diagram of the scoring test process.

Table 1 – Scoring accuracy of Custom Scoring, RAGAS and TruLens as run on five test datasets with questions run on GPT 3.5, GPT 4, GPT 4o, Claude 3 – Sonnet, and Llama 3

		Scoring Accuracy					
Model		GPT 3.5	GPT 4	GPT 4o	Claude 3 - Sonnet	Llama 3	Average
Scoring Method	Custom Scoring	96%	93%	95%	85%	91%	92%
	RAGAS	84%	82%	82%	75%	80%	80.6%
	TruLens	80%	88%	85%	74%	76%	80.6%

4.2. Formatting

The goal of the formatting test was to prove out the effectiveness of our curation process for the “Golden Document”. In the test the “Golden Document” was tested against the outputted SCTE 280 document using LlamaParse a popular framework for automated conversion of PDF documents to text. The two documents were then used as context and a set of 20 questions were taken from our 119 question set. Twenty questions were selected instead of the full test set due to time and cost of running the test.

Table 2 – SCTE 280 Curated Golden Document vs Automated LlamaParse Conversion

		Golden Document	LlamaParse
MODEL	GPT 3.5	80%	66%
	GPT 4	88%	75%
	GPT 4o	93%	83%
	Claude 3 – Sonnet	79%	63%
	Average	85%	72%

4.3. Alt-Text

The alt-text test used two versions of the SCTE 280 "Golden Document", one with alt-text included and one without alt-text. The two documents were then used as context to our RAG application. The 119 test questions were run against the two documents via our RAG application for each of the following models. The responses were then scored using our custom scoring solution. The average of the scores is shown below for each LLM.

Table 3 - Alt-text vs non-alt-text answer correctness results

		ANSWER CORRECTNESS	
		With Alt-Text	Without Alt-Text
MODEL	GPT 3.5	90%	60%
	GPT 4	95%	45%
	Claude 3 – Sonnet	85%	60%
	Llama 3	85%	50%
	Average	89%	54%

4.4. Tables

To test tables we fabricated a table based on basketball statistics to ask 20 questions about. In the first version of the test we used Table 1 from SCTE 280. In this test the answers to the question were all correct. We determined we needed a more complex table and the decision was made to fabricate our own

table. We wrote 20 questions based on the basketball statistics table and then manually graded the responses. Table 4 shows the results of this test.

Table 4 - Table format answer correctness when asked 20 complex questions on basketball statistics table

	TABLE ANSWER CORRECTNESS			
		Correct	Partially Correct	Wrong
TABLE FORMAT	AsciiDoc	19	1	0
	CSV	18	1	1
	Markdown	17	2	1
	JSON	16	3	1
	YAML	15	4	1

4.5. RAG vs non-RAG

To test the advantages of RAG vs non-RAG we enabled the ability for us to toggle RAG in our application. We then ran our set of 119 test questions on both solutions for each of the following models. The responses were then scored using our custom scoring solution. Refer to __ for a diagram of the RAG test process. Table 5 shows the results of RAG vs non-RAG.

Table 5 – RAG vs non-RAG answer correctness

		RAG	NON-RAG
MODEL	GPT 3.5	77%	66%
	GPT 4	82%	64%
	GPT 4o	83%	64%
	Claude 3 - Sonnet	83%	65%
	Average	81%	65%

5. Reliability Considerations

Reliability considerations for GAI, LLMs, and their RAG-based solutions are discussed in depth in [1]; here we cover some of those topics as they relate to the work reported in this paper.

LLMs, or for that matter any GAI, carries unique properties that make reliability considerations a greater challenge. While true these systems are for the most part software, and software reliability is a well-studied and understood topic, the generative feature stands out as a unique risk to reliable outcome. Software applications perform a function that is rather contained. Even when considered within a system and the broad use cases software can be applied to, correctly or incorrectly, most software applications exhibit the behavior of an input gives a repeatable output; not so with GAI; one cannot test with assurance, therefore. Also, LLMs might resemble search with some augmented features, but they go far beyond; they generate new content based on patterns, and that generative content is a new reliability risk.

When the output is not repeatable and the generation of content is new, how do you decide if the system meets the intended function? Let's start with what can be done with current understanding.

- Cohen's Kappa is often used as a way to determine reliability of GAIs. This method assesses inter-coder reliability, a measure of the agreement between two GAI models. That might reinforce results, but it doesn't provide assurance.
- GAI failures can be categorized as bias, and hallucinations. Others have provided further differentiation in the attempt to assess and improve on the results. But the use of LLMs and GAIs in engineering should be assessed more strongly: instead of biased we might have incomplete, and instead of hallucinations we might have incorrect.

For these reasons, we find it far better to judge the reliability of the outcome from a LLM by finding the reputable, correct response within the RAG source which supports the GAI answer. When building such a system to assess the LLM's answers, a poor result suggests a question: if we can automate the judgement that the LLM provided an incorrect or incomplete answer, can't we use that same assessment to improve on the answer?

Consider also the use of LLMs. In a creative endeavor, hallucinations might be a benefit. In engineering applications however, that may not be the case if the results break physics, math, or engineering facts. That said, a correct and creative answer can be very beneficial! The real reliability risk, however, is when the user can't tell. This risk increases when the LLM is misused, say as used in an application area outside of what it was tested for, outside of its knowledge base, or outside its general capability.

For this reason, in the future, we hope to develop solutions to assure LLMs and RAGs for use in particular application types so they can be trusted to support certain use cases with known risks.

The tools we have developed can theoretically be used to build a self-improving process too. For example, by developing standards, specifications, and technical documents using a real time RAG generator and LLM, the experts can test their own output in near real time. By giving the resulting LLM with the draft RAG embedded, the experts can give questions to the LLM and determine the quality of the draft document, thereby clarifying what is in the document and improving in the output, for human readers, LLMs, and the tools built from them.

Ultimately, LLMs including the ones we have built will need to build trust before they will be used in mission critical applications, including customer impacting uses. As we improve on the reliability of these solutions,

Research and development will continue. Research and development ultimately is about making experimental results reliable at scale. We can scale LLMs, so we'll keep working on making them more reliable. To do that, we also have to improve on how to judge reliable output.

6. Security Considerations For RAG

RAG is a revolutionary tool for finding and understanding information. However, it does introduce several new privacy and security concerns. Some of these, such as prompt injection and membership inference, arise from the use of LLMs in general, however RAG introduces specific security issues, given their access to large knowledge bases and potential to generate sensitive and internal information. There are four main areas that we consider regarding the security and privacy of RAG-based systems: securing the data used, securing the query inputs and retrieval process, and securing the output and generation step .

6.1. Privacy and Security of the Data

A critical aspect of securing RAG systems is to safeguard the retrieval document corpus and knowledge base storage by protecting sensitive information from unauthorized access. Implementing access controls is a primary component of this, however many current vector databases lack fine-grained access controls. If access control cannot be done at the data access layer the application layer must handle it. One method of doing this is to separate embedding stores based on roles, and then implement role-based access control (RBAC) at the application layer allowing access to specific embedding databases based on the role of the requestor. This will help prevent unauthorized access situations, for example, avoiding the mixing of customer data and HR data in the same vector store.

Ensuring the integrity of knowledge stores is also important and a hash should be run and stored each time a vector database is changed. When including third-party inputs outside of the organization's direct control, such as results from webpages, input sanitization checks should be performed on this data.

It is important to note that vector databases can be semantically reversed, text representing the meaning of the knowledge corpus can be extracted, potentially exposing sensitive information. Encryption for stored data should be applied to both source documents and vector/graph stores. If customer data is being collected, then ensuring data privacy and compliance with the growing number of privacy regulations like GDPR and CCPA is also essential. Pre-parsing knowledge corpus documents and removing personally identifiable information prior to embedding is recommended. These security measures collectively help maintain the integrity, confidentiality and privacy of data within RAG systems.

6.2. Query and Retrieval Security

Query and retrieval security focuses on safeguarding the process of querying and retrieving information. This involves protecting user queries from interception or manipulation through TLS, which should be implemented between the user and frontend, and for any frontend calls back to vector stores and LLMs.

Implementing authentication and authorization for users and other entities accessing the system is another key aspect of query and retrieval security. This will ensure that no unauthorized parties can query the system and help prevent membership inference attacks. Additionally, queries should have an upper limit on the number of characters submitted, and input sanitization should be conducted to detect and prevent injection attacks. A context-specific semantic check is also recommended to ensure that queries are within a reasonable semantic similarity to the underlying source data. For instance, if a query about car

transmissions is made against a DOCSIS specification, the system should prompt the user to retry the query.

6.3. Generation and Output Security

Generation security focuses on securing the generation process and its outputs. Ensuring that generated content doesn't reveal sensitive information by requiring a post-processing filtering step that can range from simple keyword searches to a secondary LLM inspection to remove results that contain sensitive or harmful information. This is similar to the method we used to score the initial output results for accuracy.

Securing the model itself from potential attacks or unauthorized modifications is equally vital. This involves signing and authenticating model weights. These signatures should be checked at each running instance. For models that undergo training or fine-tuning, companies should, to the extent possible, sign the training data and verify its signature each time the model is trained. These security measures will help maintain the integrity and authenticity of the generation process in RAG systems.

7. Future Work

Progress and change in AI is evolving very fast. There are three main areas we see as next steps to improve the current work: advanced retrieval methods, multi-modal retrieval/generation, and agentic workflows. As we improve on these three fronts, we will also improve on reliabilty and security.

7.1. Advanced RAG and Knowledge Graphs

Retrieving the most relevant information from a knowledge corpus is the primary goal of RAG, however there is always room for improvement. In future work we will explore advanced techniques for RAG including adaptive chunk size optimization and knowledge graphs. In adaptive chunk size optimization, the amount of text returned can vary based on the query type. For example, factual queries require smaller context and focus on precise results e.g. “What are the modulation techniques in DOCSIS 3.1”. More open-ended questions require more context and focus on more generalized retrieval e.g. “What are the differences in modulation techniques in DOCSIS 3.1 and 4.0 and how do they improve performance in DOCSIS 4.0?” We will also explore additional chunk sizing techniques based on hierarchy windows that take advantage of the natural hierarchy in technical documents such as sections, and chapters as well as semantic window techniques that preserve the natural document structure.

Knowledge graphs are another technique that can be used to enhance retrieval by providing additional context over the knowledge base. Knowledge graphs break information into a network of nodes and edges, where the nodes are things like people, places, concepts. Edges represent the relationships between the nodes. A simple example of this in DOCSIS would represent latency and user experience as nodes and an edge of “impacts” representing the connecting relationship. This could be further refined as

nodes like, high latency, poor user experience, lag. and the relationship would be “causes” or “results in”. This is shown in Figure 7 below.

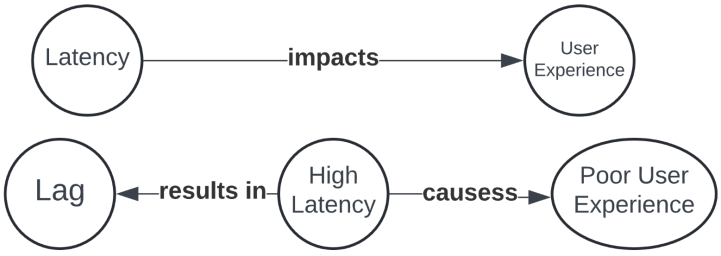

Figure 7 - Knowledge Graph Example

7.2. Multi-Modal Sessions

There is a lot of information that does not reside in text documents. Future work will expand from query/answer chat sessions to full multimodal sessions that include both input and output of images, audio, voice, and video. We foresee that future LLMs will be able to converse in natural voice in any language with humans in real-time in the field, by helping technicians to diagnose problems by examining spectral analysis along with visual input from the physical plant.

7.3. Agent-Based Workflows

Agent-based workflows are built upon LLMs by using software agents to perform tasks, make decisions, and interact with other systems or humans. An AI agent is built by adding several contextual prompts to an LLM, these prompts give it focus and can let it take on a persona with specific skills that include data collection (retrieval), analysis, diagnosis, solution generation (including code), planning, execution (including code). In a future work we will examine breaking out agents to do specific retrieval operations, content moderation operations, and more.

8. Conclusion

The effort of creating a CableLabs domain expert LLM has proven to be a formidable adversary. Through experimentation, iterative advances and solution validation we have been able to show iterative progress and positive results. The advances we have made in document curation by carefully choosing the format of the document, tables, figures and addition of alt-text have dramatically improve the outputs of our RAG application.

As part of our journey, we have generated a new custom scoring solution, which has shown extremely positive performance. With this scoring solution we have enabled ourselves to automate the testing of various aspects of the RAG pipeline and iteratively prove the effectiveness of our RAG application. The custom scoring solution has been instrumental to the success of progress and validation of our findings.

We have shown RAG to be an effective solution for the mitigation of issues with LLMs including hallucination, false information and lack of accountability. In doing so we have laid the foundation for a future in which we can utilize LLMs to solve some of the more time intensive tasks in the cable industry. It is our hope the techniques and knowledge we have gained through our trials with RAG can be applied not only to answer questions relating to domain expertise, but also be used to help field technicians troubleshoot in real-time.

We look forward to the future of RAG and LLMs. In the next generation of our work, we hope to synchronize with the current direction of the industry. With the help of agents, knowledge graphs, multi-modal models we plan to improve our RAG pipeline, ultimately incorporating RAG into larger applications as part of multi agent architectures. The possibilities are endless for the use of LLMs and RAG in building the applications of the future.

The raw data proved too large to include in this document. For questions and access to the code and data please contact the authors.

Abbreviations

AP	access point
bps	bits per second
FEC	forward error correction
HD	high definition
Hz	hertz
K	kelvin
LLM	large language model
RAG	retrieval augmented generation
SCTE	Society of Cable Telecommunications Engineers

Bibliography & References

Include an annotated bibliography of key resources providing additional background information on your topic.

[1] Rupe, J., "Reliability of generative artificial intelligence," IEEE Reliability Magazine, September 2024.

[2] Network Operations Subcommittee, SCTE 280 - Understanding and Troubleshooting Cable RF Spectrum. Society of Cable Telecommunications Engineers, Inc., 2022.

[3] OpenAi, "ChatGPT," chatgpt.com, 2024. https://chatgpt.com

[4] "Danswer - Open Source Workplace Search," www.danswer.ai. https://www.danswer.ai

[5] "h2oGPT | H2O.ai," h2o.ai. https://gpt.h2o.ai

[6] "open-webui/open-webui," GitHub. https://github.com/open-webui/open-webui

[7] "AnythingLLM | The ultimate AI business intelligence tool," useanything.com. https://anythingllm.com

[8] "Empower Your Enterprise With AI," bionic-gpt.com. https://bionic-gpt.com

[9] "AutoGPT: the heart of the open-source agent ecosystem," GitHub, Sep. 26, 2023. https://github.com/Significant-Gravitas/AutoGPT

[10] "LangChain," www.langchain.com. https://www.langchain.com

[11] "Ragas," ragas.io. https://ragas.io

[12] TruEra, "TruLens," www.trulens.org. https://www.trulens.org

[13] "Pandoc" pandoc.org. https://pandoc.org

[14] "LlamaIndex, Data Framework for LLM Applications," www.llamaindex.ai. https://www.llamaindex.ai

[15] CableLabs, Data-Over-Cable Service Interface Specifications DOCSIS® 4.0 - MAC and Upper Layer Protocols Interface Specification. 2024.

Appendix

9. Appendix A – First 10 SCTE 280 Test Questions

The first ten questions from the SCTE 280 test question set are provides as an example reference for the questions asked. The questions in the full 119 test set include questions about SCTE 280 ranging from simple to more complex.

Table 6 - First 10 SCTE 280 Test Questions

#	Question	Expected Answer
1	What kind of devices does SCTE 280 focus on?	This document covers devices that work with cable TV and internet (DOCSIS 3.0 and above) that have a feature called "full band capture" to check signal quality. It does not include devices like MoCA or Wi-Fi, which work differently.
2	Who is the intended audience of SCTE 280?	This material applies to field technicians, teams doing analysis and device repair, software designers, and systems engineers.
3	How is SCTE 280 useful to its audience?	Cable TV and internet field technicians and engineers can use this document to understand cable signal performance better.
4	What are the goals of SCTE 280?	This document aims to help technicians fix problems faster and more accurately by providing a consistent way to interpret and troubleshoot cable signal issues specific to coaxial cable networks.
5	What is radio frequency (RF)?	Radio frequency is the portion of the electromagnetic spectrum from a few kilohertz to about 300 GHz. RF can also be described as a rate of oscillation within the 3 kHz to 300 GHz range.
6	What is Direct current (DC)?	Direct current, abbreviated DC, is an electric current that is unidirectional, as a result of a voltage source whose output maintains the same polarity.
7	Can you provide an example of Direct current (DC)?	An example is the output of a flashlight battery.
8	Is DC a type of RF?	DC is not RF, nor is it part of the electromagnetic spectrum.
9	What is Alternating current (AC)?	Alternating current, abbreviated AC, is an electric current that

		periodically reverses or alternates in direction, as a result of a voltage source whose output periodically reverses or alternates in polarity.
10	Can you provide an example of AC?	Examples include AC from a household electrical outlet and RF signals.

10. Appendix B – First 10 GPT 4o Scoring Test Results

The first ten results from running the scoring test using GPT 4o as the LLM model. The entire results are two large to include in this document and are available upon request. This is a sample of the results.

Table 7 - First 10 GPT 4o Scoring Test Results

		Expert Score		Custom Scoring		RAGAS Scoring		TruLens Scoring	
Question #	**Answer**	**Expert Score**	**Expert Score Percent**	**Answer Correctness**	**Scoring Accuracy**	**Answer Correctness**	**Scoring Accuracy**	**Answer Correctness**	**Scoring Accuracy**
1	SCTE 280 focuses on DOCSIS customer premise equipment (CPE).	5	50%	50%	100%	58%	92%	100%	50%
2	The intended audience of SCTE 280 is primarily field-facing technicians, but the document is also useful for analysis and repair teams. It may also be useful for software designers and systems engineers.	10	100%	100%	100%	67%	67%	100%	100%
3	SCTE 280 is useful to its audience by providing specifications, test methods, and procedures that promote uniformity of product, interoperability, interchangeability, best practices, and the long-term reliability of broadband communications facilities. It enables cable modems to report RF spectrum data, allowing operators to observe RF performance in unprecedented ways. Additionally, it helps in identifying and troubleshooting common issues in the RF spectrum, thereby aiding in effective network operations and maintenance.	7	70%	75%	95%	86%	84%	100%	70%

4	The goals of SCTE 280 are to provide specifications, test methods, and procedures that promote uniformity of product, interoperability, interchangeability, best practices, and the long-term reliability of broadband communications facilities.	5	50%	0%	50%	20%	70%	90%	60%
5	Radio frequency (RF) is that portion of the electromagnetic spectrum ranging from a few kilohertz to about 300 GHz. It is a form of electromagnetic radiation that propagates through free space at the speed of light and is made of photons. RF energy can be detected and measured with specialized test equipment and can convey information by varying its amplitude, frequency, or phase. It can be transmitted via conductors like coaxial cable, over-the-air, or through the vacuum of space.	8	80%	100%	80%	84%	96%	100%	80%
6	Direct current (DC) is an electric current that is unidirectional, as a result of a voltage source whose output maintains the same polarity. An example of DC is the output of a flashlight battery.	10	100%	100%	100%	67%	67%	100%	100%
7	An example of Direct current (DC) is the output of a flashlight battery.	10	100%	100%	100%	98%	98%	100%	100%
8	No, Direct current (DC) is not a type of radio frequency (RF).	8	80%	75%	95%	60%	80%	100%	80%
9	Alternating current (AC) is an electric current that periodically reverses or alternates in direction, as a result of a voltage source whose output periodically reverses or alternates in polarity. Examples include AC from a household electrical outlet and RF signals.	10	100%	100%	100%	85%	85%	100%	100%
10	An example of Alternating Current (AC) is the electric current from a household electrical outlet.	10	100%	75%	75%	72%	72%	100%	100%

11. Appendix C – LlamaParse Example Figure Output

Figure 9 shows why there is a need for alt-text and appropriate figure formatting. The figure was converted to text using one of the most popular frameworks, LlamaParse. This is one example out of many showing why automated conversion fails and the need for manual alt-text or text based figure formats such as Mermaid.

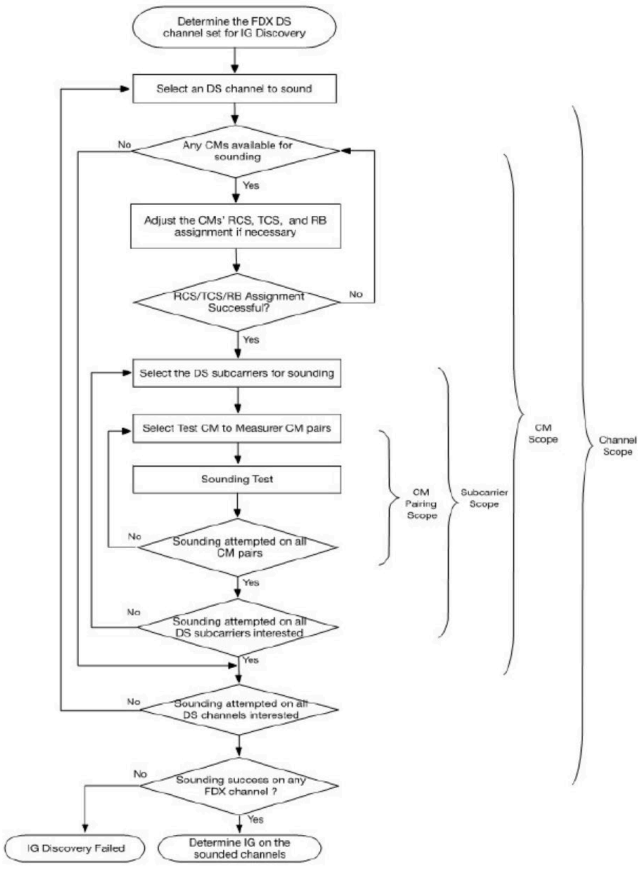

Figure 8 - Figure 267 from MULPIv4.0-N-24.2370-3 [15]

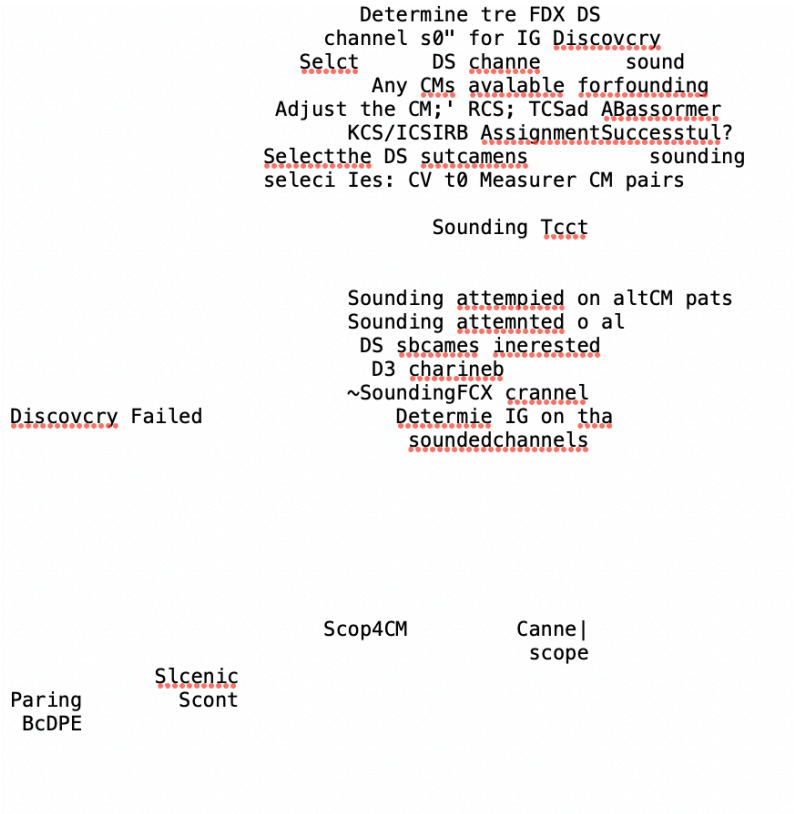

Figure 9 - LlamaParse output from Figure 267

12. Appendix D – Basketball Statistics Table and Example Questions

Below in Table 8 is shown the basketball statistics table fabricated for table testing. After the figure are the list of questions run on the table.

Table 8 - Basketball statistics table

Basketball Data Table							
Player	Team	Position	Points Per Game	Rebounds Per Game	Assists Per Game	Steals Per Game	Blocks Per Game
LeBron James	Los Angeles Lakers	Forward	30.3	8.2	6.2	1.1	0.6
Kevin Durant	Phoenix Suns	Forward	29.1	6.8	5.3	0.8	0.8
Giannis Antetokounmpo	Milwaukee Bucks	Forward	31.1	11.8	5.7	1	1.1
Stephen Curry	Golden State Warriors	Guard	29.4	6.1	6.4	1.5	0.4
Luka Doncic	Dallas Mavericks	Guard	28.4	9.1	8.7	1.2	0.5
Joel Embiid	Philadelphia 76ers	Center	33.1	10.2	4.2	0.8	1.7
Nikola Jokic	Denver Nuggets	Center	24.5	11.8	9.8	1.5	0.6
Jayson Tatum	Boston Celtics	Forward	30.1	8	4.6	1	0.5
Ja Morant	Memphis Grizzlies	Guard	27.1	5.9	8.1	1.7	0.3
Damian Lillard	Portland Trail Blazers	Guard	24	4.2	7.3	1.1	0.3
Donovan Mitchell	Cleveland Cavaliers	Guard	28.3	3.9	4.8	1.5	0.3
Paul George	Los Angeles Clippers	Forward	23.8	6.5	5.1	1.1	0.4
Kyrie Irving	Dallas Mavericks	Guard	27	5.1	6.2	1.4	0.6
Anthony Davis	Los Angeles Lakers	Center	21.9	7.9	3.1	1	2
Karl-Anthony Towns	Minnesota Timberwolves	Center	24.6	9.8	5.3	0.9	1.1

1. Basic Information: "How many players are listed in the table?"

2. Team Representation: "Which team has the most players represented in the table?"

3. Average Calculation: "What is the average points per game for all players listed?"

4. Data Extrema: "What is the highest points per game recorded in the table?"

5. Position-Specific Retrieval: "Who has the most assists per game among the forwards?"

6. Data Aggregation: "Which player has the highest rebound per game average?"

7. Count by Category: "What is the total number of players listed who play the guard position?"

8. Category Average: "What is the average steals per game for all centers?"

9. Data Extrema (2): "Which player has the lowest blocks per game average?"

10. Combined Data Retrieval: "List the players who average more than 25 points per game and more than 6 assists per game."

11. Specific Data Retrieval (Team): "List all players on the Los Angeles Lakers."

12. Data Aggregation (Team): "Which team has the player with the highest rebounds per game?"

13. Filtered Average: "What is the average points per game for all players on teams starting with the letter "M"?"

14. Team Comparison: "Which team has the highest average points per game among the players listed?"

15. Conditional Retrieval: "List all players who have a higher assists per game average than their steals per game average."

16. Table Transformation: "Create a new table showing only the top 5 players ranked by points per game."

17. Simple Calculation: "If a player averaged 10 more points per game, what would their new points per game average be? Create a new table just with Player and new Points Per Game."

18. Sorting and Ordering: "If the table were sorted by assists per game in descending order, who would be in the top 3 positions?"

19. Category Averages: "Can you calculate the average points, rebounds, assists, steals, and blocks per game for each position (forward, guard, center)?"

20. Hypothetical Data Integration: "Imagine a new player, named ‘Test Player’ is added to the table with 35 points, 12 rebounds, 5 assists, 1 steal, and 8 blocks. Create a new table adding this player and showing only the top 5 players ranked by blocks per game."