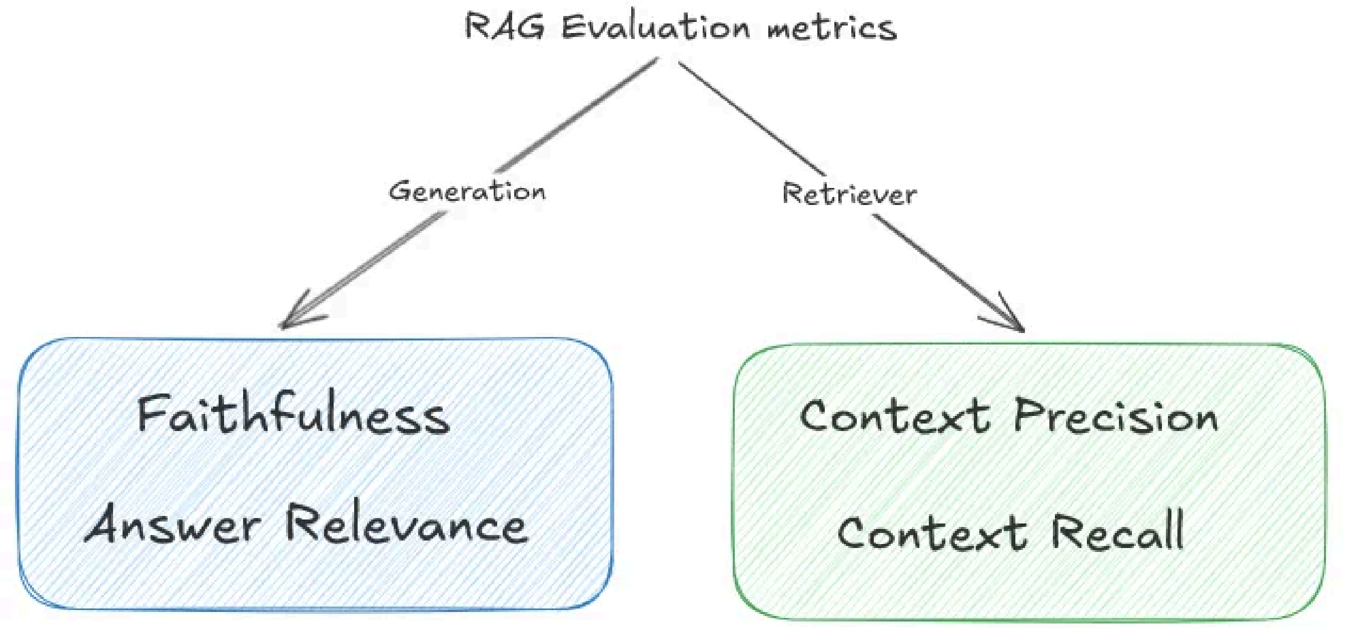
RAG Evaluation Metrics Explained: A Complete Guide

10 min read · Aug 11, 2024

Mohamed EL HARCHAOUI Follow

RAG (Retrieval-Augmented Generation) enhances generative language models by integrating information retrieval techniques to address the issue of producing plausible but sometimes factually incorrect responses. By retrieving relevant information from external sources, RAG reduces the likelihood of factual errors (hallucinations), leading to more reliable and accurate content.

RAG combines two main components: **Retrieval** and **Generation**. The Retrieval component involves *indexing* and *searching* external knowledge sources to find relevant information, while the **Generation** component uses this retrieved data to create coherent and contextually appropriate responses. This approach makes RAG a significant advancement in developing more intelligent and versatile language models.



RAG evaluation metrics overview

The evaluation of a RAG system can be divided into two parts same as RAG system: one for evaluating the **retriever** component and another for evaluating the **generation** component. The former assesses the quality of the entire pipeline up to the retrieved context, while the latter evaluates the generated content based on the retrieved information.

Let's start by the retriever part :

Evaluating the Retriever Component

The retriever is a crucial component in a RAG system, contributing to approximately 90% of the systems overall quality and value.

To evaluate the quality of the retriever, we need to examine the retrieved context based on a given query. But how do we determine if the retrieved context is accurate and relevant? For this, we use a ground truth, a reference dataset that serves as a benchmark to evaluate the quality of the retrieved context. we will look at this in details in the following.

Retriever part has two major metrics : **Context precision** and **context recall**. I will give a detailed explanation of each one.

Context precision :

When you make a query on a search engine like Google, you typically expect to find what you're looking for in the first few links. This is precisely what context precision measures. It evaluates whether the relevant ground truth contexts are ranked higher in the list of retrieved contexts.

This metric is computed using context and the ground truth context, it formula is as below.

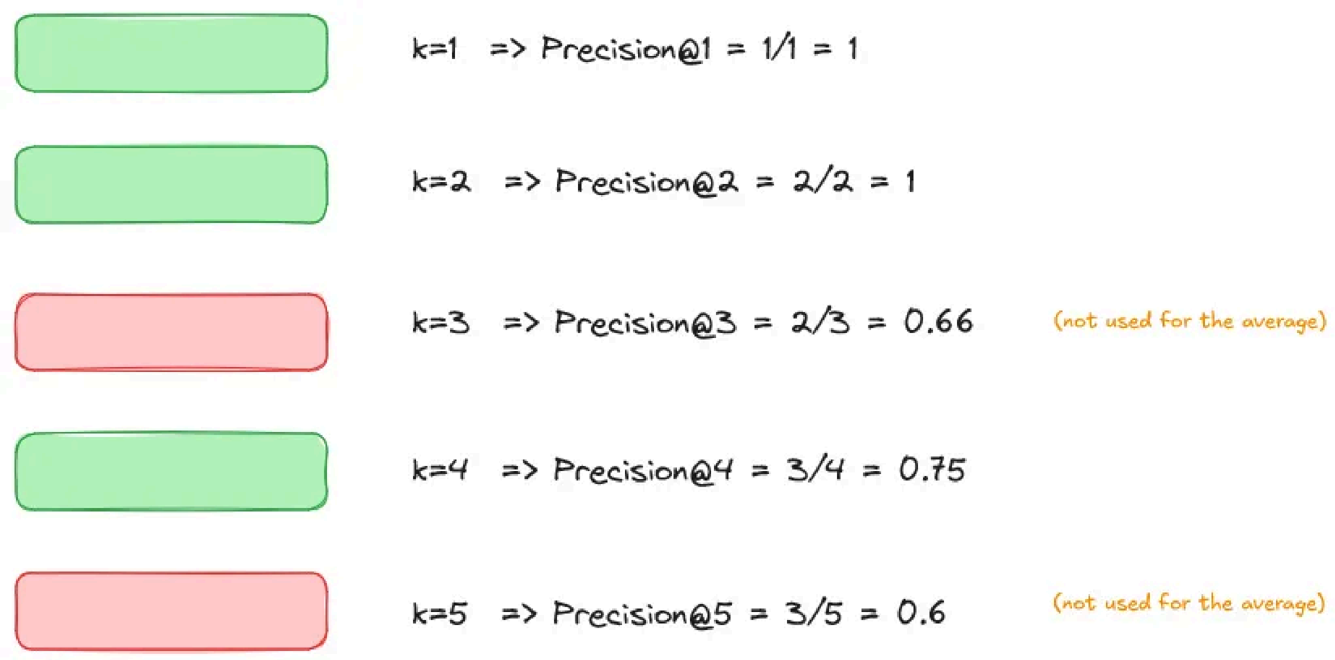
$$\text{Context Precision@k} = \frac{\Sigma\,(\text{Precision@k} * \text{Vk})}{\text{Number of relevant items in top K}}$$

Where Precision at K is calculated as below:

$$\text{Precision@k} = \frac{\text{true positives@k}}{(\text{true positives@k}) + (\text{false positives@k})}$$

Let’s use illustrations and examples for a clearer understanding.

Suppose your system have retrieved 5 chunk/context, from those context, three of them are ground truth as shown below. Following the formulas of Precision up to K elements you can calculate them as below:



🟩 k=1 => Precision@1 = 1/1 = 1

🟩 k=2 => Precision@2 = 2/2 = 1

🟥 k=3 => Precision@3 = 2/3 = 0.66 (not used for the average)

🟩 k=4 => Precision@4 = 3/4 = 0.75

🟥 k=5 => Precision@5 = 3/5 = 0.6 (not used for the average)

Now that you have calculated the precision@K, you will use only the calculated precision at ground truth K to compute the Context Precision as follows:

$$\text{Context Precision@k} = \frac{\Sigma\ (\text{Precision@k} * \text{Vk})}{\text{Number of relevant items in top K}} = (1+1+0.75)/3 = 0.92$$

Now let’s take a real world example:

Question: What is the boiling point of water at sea level and what is its chemical formula?

Ground truth: The boiling point of water at sea level is 100°C, and its chemical formula is H_2O.

High context precision: [“Water, with the chemical formula H_2O, boils at 100°C at sea level. It is a vital substance for all known forms of life, covering about 71% of Earth’s surface.”, “The boiling point can vary with altitude, but at sea level, it consistently boils at 100°C, a key property of this essential compound.”]

Low context precision: [“Water is essential for life and covers a significant portion of Earth’s surface. It has a unique ability to dissolve many substances, making it a versatile solvent.”, “Water, with the chemical formula H_2O, boils at 100°C at sea level. It is a vital substance for all known forms of life.”]

Let’s calculate the context precision for the **Low context** example :

Step 1: check relevance

There is tow chunks, the first one is not relevant to ground truth, the second one is relevant

Step 2: Calculate precision at K

Precision@1 = 0/1 and Precision@2= 1/2 = 0.5

Step 3: Calculate the mean of precision@k :

Context Precision : (0+0.5)/1 = 0.5

Context Recall

Context recall assesses the accuracy of the retrieved context by comparing the facts and claims it contains against the ground truth. Essentially, it indicates the proportion of claims from the ground truth that your context successfully captures.

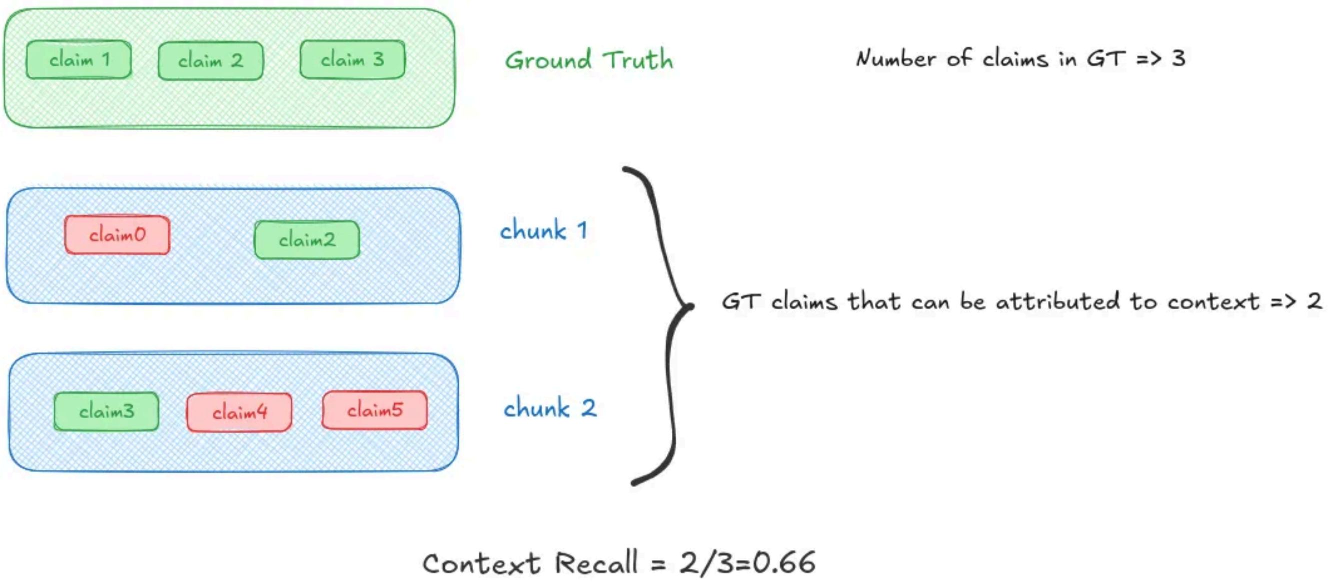
The value ranges from 0 to 1, where a higher value indicates better results. To calculate this value, the ground truth is broken down into a set of claims. Each claim is then analyzed to determine whether it is present in the retrieved context.

A perfect context recall occurs when all claims from the ground truth are found within the retrieved context.

It's calculation formula is as below :

$$\text{Context Recall} = \frac{\text{GT claims that can be attributed to context}}{\text{Number of claims in GT}}$$

Here is an illustration to clarify this measure :


claim 1
claim 2
claim 3
Ground Truth
Number of claims in GT => 3
claim0
claim2
chunk 1
GT claims that can be attributed to context => 2
claim3
claim4
claim5
chunk 2
Context Recall = 2/3=0.66

Let's take a real world example

if we apply this to our previous example about chemical water specification. For the low context we can do :

Ground truth: The boiling point of water at sea level is 100°C, and its chemical formula is H_2O.

Low context recall: [“Water is essential for life and covers a significant portion of Earth’s surface. It has a unique ability to dissolve many substances, making it a versatile solvent.”, “Water, with the chemical formula H_2O. It is a vital substance for all known forms of life.”]

Step 1 : Break down ground truth to statements

Statement 1 : “The boiling point of water at sea level is 100°C”

Statement 2 : “ Water chemical formula is H_2O.”

Step 2 : For each GT Statement, check if it’s can be attributed to the retrieved context

Statement 1: No

Statement 2: Yes

Step 3 : Use the formula to compute the metric:

Context recall : 1/2 = 0.5

Evaluating the generation component:

The generation component uses LLMs to generate responses to users; it takes as input the context and the question and tries to provide a useful answer to the user. LLMs are subject to hallucinations and may generate some fictional facts that do not exist or are not related to the provided context. It’s very important to evaluate the quality of what was generated.

In this part we can have several metrics, in this blog we will explain two of the most used metrics, which are the Faithfulness and the Answer relevance.

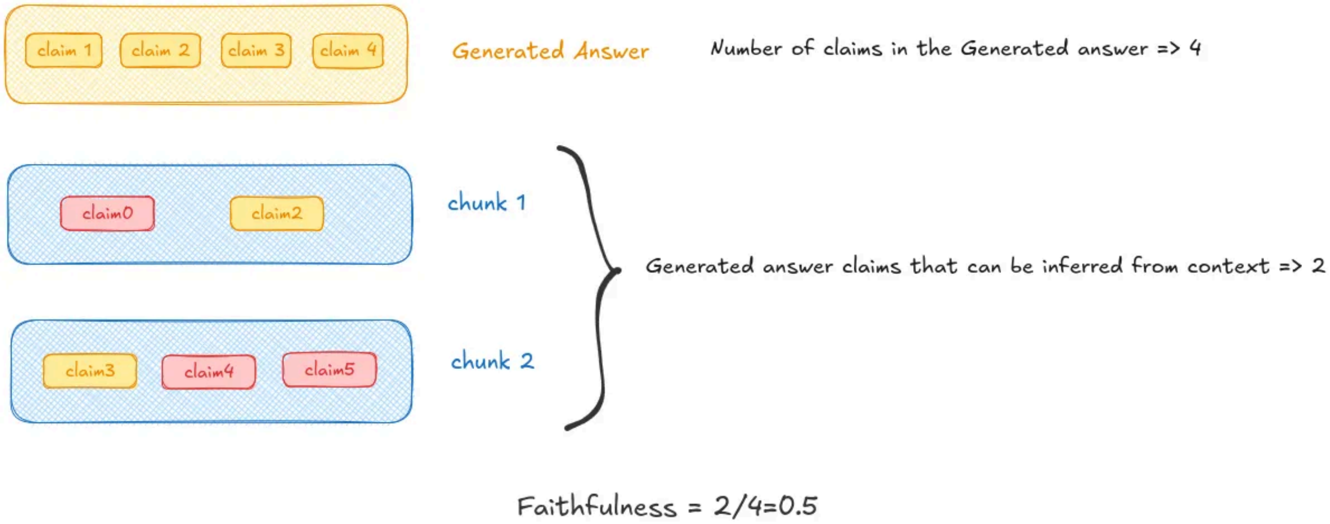
Faithfulness

This metric measure how the LLM answer is faithful to the provided context, does it respect what was given as input or not. Its considered as faithful if the claims made in the answer can be extracted from the provided context. To calculate it, we start by extracting all claims from the LLM provided answer first. Then for each claim we check if this one claim can be inferred from the retrieved context. It value range from 0 to 1. Higher is better.

It’s formula can be written as below :

$$\text{Faithfulness} = \frac{|\text{Number of claims in the answer that can be inferred from the given context}|}{\text{Number of claims in the answer}}$$

Here is a visual illustration of this:



Now lets give a real world example :

Question: What was the outcome of the Battle of Waterloo?

Context: The Battle of Waterloo, fought on 18 June 1815, marked the final defeat of Napoleon Bonaparte. The battle took place near Waterloo in present-day Belgium, and resulted in a decisive victory for the Seventh Coalition, effectively ending the Napoleonic Wars.

High faithfulness answer: The Battle of Waterloo resulted in a decisive victory for the Seventh Coalition on 18 June 1815, marking the final defeat of Napoleon Bonaparte.

Low faithfulness answer: The Battle of Waterloo resulted in a decisive victory for the French forces on 18 June 1815, ending the Napoleonic Wars.

For the **low faithfulness answer** can calculate the metric as below:

Step 1 : Break down the generated answer to claims

Claim 1 : “The Battle of Waterloo resulted in a decisive victory for the French forces”

Claim 2 : “The Battle of Waterloo took place on June 18, 1815.”

Step 2 : For each answer claim, check if it’s can be inferred from the given context

Claim 1: No

Claim 2: Yes

Step 3 : Use the formula to compute the faithfulness:

Faithfulness = 1/2 = 0.5

Answer Relevance

This metric measure the quality of the generated answer given the user query, how pertinent is the answer with respect the the user question. To assess this we need to know if the answer is complete or not, does it contain redundant information ?

To calculate this metric, we generate N question based on the answer, does questions should be normally similar the the original question if the provided answer is relevant to the original question, if not they will be different. To compare the N generated question, we use cosine or dot product vector similarity operators. The value should range between 0 and 1.

The formula for determining answer relevance is as follows:

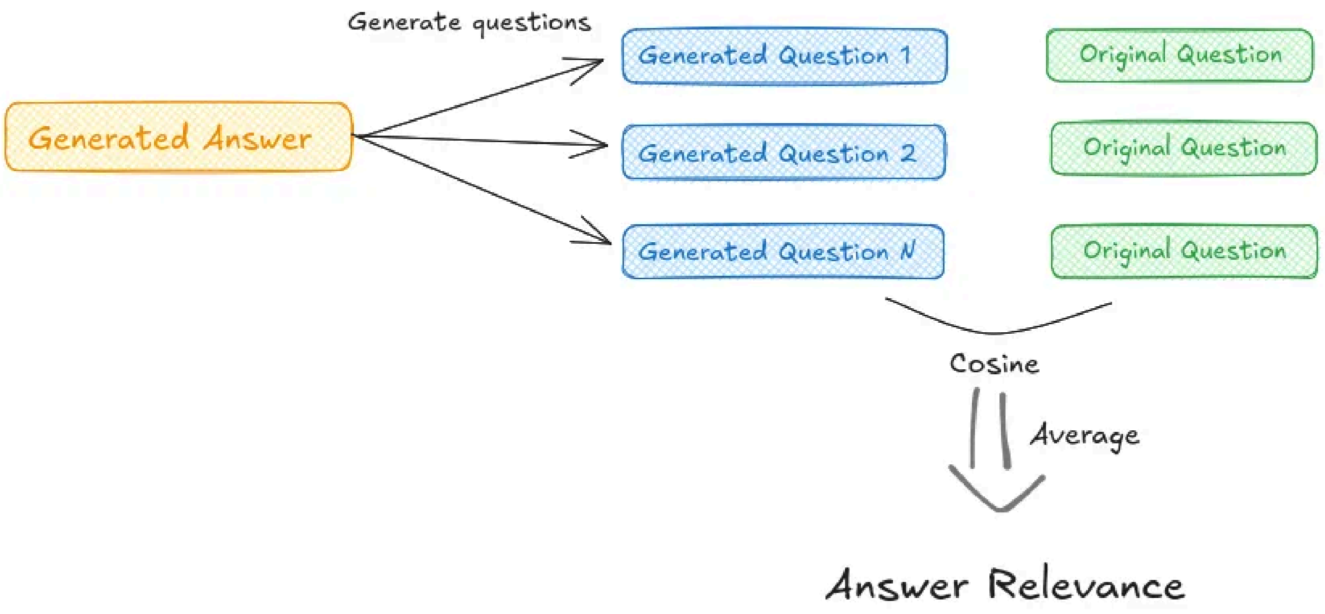
$$\text{Answer Relevance} = \frac{\sum\left[\text{Cos}(\text{Embedding\_Gi}, \text{Embedding\_O})\right]}{N}$$

Where :

- **Embedding_Gi** : is the embedding of the generated question i
- **Embedding_O** : is the embedding of the original question
- **N** : is the number of generated question, generally between 3 to 4

Note : Cosine similarity scores range from -1 to 1, indicating that in extreme cases, the value may fall anywhere within this interval.

Below illustration show the process of building this metric :



Let's use a real-world example to explain this :

Question: What is the chemical formula for water, and what are its primary components?

Low relevance answer: Water is essential for life and is found in oceans, rivers, and lakes.

High relevance answer: The chemical formula for water is H2O, and its primary components are hydrogen and oxygen.

Given that example let's calculate the answer relevance for the "low relevance answer". To do that, we can follow the same process as illustrated above.

Step 1: Use LLM to generate 3 questions based on the low relevance answer. here are example of 3 generated questions:

1. Why is water considered essential for life?
2. What are the primary sources of water found on Earth?
3. How do oceans, rivers, and lakes contribute to the water cycle?

Step 2 : Use embedding model to generate embedding vector for generated question and the original question.

Step 3 : Compute average cosine similarity between the embedding of the generated and the original question.

the cosine score i got for this example is : tensor([0.5408, 0.5868, 0.4486])

so the Relevance answer is 0.52 which is not good !

That’s it! I hope that you have a clear comprehension of those metrics and how to calculate them.

What’s Next ?

There are more other metrics that are used to evaluate RAG components like :

- Context entity recall
- Context utilization
- Context Relevancy
- Hallucination

If you want to me to cover them also let me know in comment/response.

Here are some references and papers if you want to delve deeper into this matter:

- **RAGAS** : https://arxiv.org/pdf/2309.15217
- **A Unified Evaluation Process of RAG Paper** : https://arxiv.org/pdf/2405.07437
- **DomainRAG** : https://arxiv.org/abs/2406.05654v2
- **FeB4RAG** : https://arxiv.org/abs/2402.11891v1

Some frameworks to help you evaluate your RAG application:

DeepEval

DeepEval is a comprehensive evaluation framework that supports a wide range of metrics for both RAG and fine-tuning use cases. It offers over 14 evaluation metrics. DeepEval is known for its modular components, making it easy to integrate and customize. It also supports Pytest integration for treating evaluations as unit tests and provides a hosted platform for real-time evaluations.

RAGAs

RAGAs is specifically designed for evaluating RAG pipelines. While similar to DeepEval, RAGAs is noted for its simplicity and ease of use, although it lacks some of the advanced features and flexibility found in DeepEva

MLFlow LLM Evaluate

MLFlow LLM Evaluate is a versatile framework that supports RAG evaluation as well as other LLM evaluation tasks. It is appreciated for its intuitive developer experience and modular design, allowing for seamless integration into existing evaluation pipelines. MLFlow is particularly useful for those who need a straightforward and flexible evaluation tool

Deepchecks

Deepchecks is another open-source framework that, while not exclusively focused on RAG, offers robust evaluation capabilities for LLMs. It is more oriented towards evaluating the LLM itself rather than the entire RAG pipeline. Deepchecks is known for its strong visualization and dashboard capabilities, which help users to easily interpret evaluation results

Arize AI Phoenix

Arize AI Phoenix is an evaluation framework that, although less focused on RAG specifically, provides valuable tools for assessing LLM performance. It is particularly useful for users who need comprehensive evaluation and monitoring solutions for their AI models

> *My final thought : “**Evaluation and creation are separate skills.**”*

What I mean is that humans can evaluate the quality of something without necessarily having the skills or intelligence to create it themselves. In other words, the ability to assess or judge the quality of something doesn’t require the same level of expertise or ability needed to produce it at a high standard.

This distinction is important because it highlights how critical thinking and judgment can operate independently of creative or technical expertise. For example, many people can recognize a beautiful piece of art, an effective piece of writing, or a well-engineered product without having the skills to create such things themselves.

Llm Llm Evaluation Rag Rag Evaluation Ragas