

Introducing REMi, the first ever open-source RAG evaluation model

Retrieval Augmented Generation (RAG) methods have become the go-to technology for document search, utilizing the latest advancements in Large Language Models (LLMs) and Deep Learning. However, evaluating these complex systems is challenging due to the intricate nature of LLM outputs. At Nuclia, we've tackled this with **REMi**, our efficient open-source LLM fine-tuned specifically for RAG evaluation. Additionally, we've developed [nuclia-eval](#), an open-source library that simplifies the assessment of RAG pipelines using REMi.

This article will provide the reader with a comprehensive understanding of the state-of-the-art metrics for evaluating RAG systems and showcase our approach with REMi and nuclia-eval.

Overview

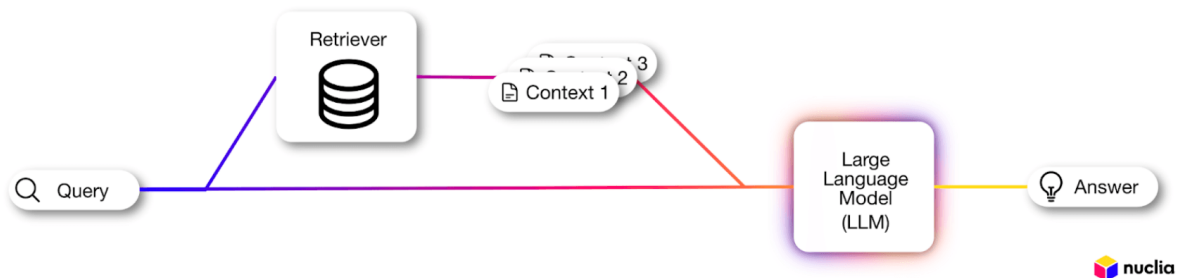
What is RAG?

Retrieval Augmented Generation (RAG) is a procedure designed to answer user questions, also called queries. It combines an information retrieval system with text generation, typically using a Large Language Model (LLM) like ChatGPT.

At **Nuclia**, we provide RAG-as-a-Service solutions, which allows clients to leverage the power of complex RAG pipelines without the need to develop their own solution and manage the underlying infrastructure.

In its simplest form, RAGs involve two steps:

1. **Retrieval:** Information relevant to the user query is retrieved from a document (or knowledge) database. It is typically done with traditional search techniques such as BM25, or with more modern techniques involving *embeddings*, obtained from a deep learning model.
2. **Generation:** The retrieved information, combined with the user's query, is used by the LLM to generate a response.



Simple visual representation of a RAG pipeline

This method leverages the power of LLMs while providing a means to control and enhance the information used to generate responses. It is particularly useful for providing answers with data not commonly found in the LLM's training data, such as specific domain, company, or product knowledge.

The main inputs/outputs in the RAG pipeline are:

- **Query:** The user's question, which the model will try to answer
- **Context Pieces:** The information retrieved by the retrieval step, which aims to be relevant to the user's query.
- **Answer:** The response generated by the language model after receiving the query and context pieces.

Current evaluation metrics for RAG

Evaluating RAG pipelines is challenging due to the complexity of the pipeline and the involvement of multiple models, including an LLM, which is inherently difficult to evaluate.

When a dataset with ground truth answers and relevant context pieces is available, the generated answers and retrieved contexts can be compared to

their ground truth counterparts using classical NLP metrics. However, in real-world scenarios without ground truth, reference-free metrics are required. To address this, the community has proposed metrics that use LLMs as evaluators.

These implementations are notably found in the libraries **DeepEval**, **RAGAS** and **TruLens**. They rely on general-purpose LLMs, often from external providers, to evaluate the quality of the generated answers.

Current approaches to evaluate RAG follow two main paths:

RAGAS framework:

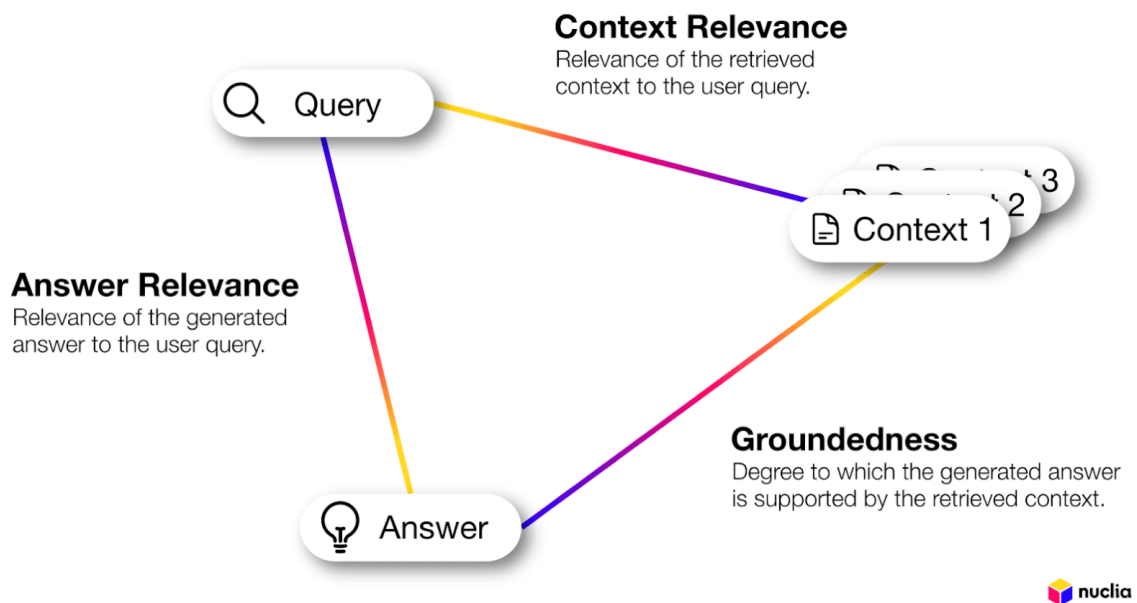
Proposes four metrics to holistically evaluate the quality of the RAG experience. **RAGAS** and **DeepEval** contain implementations of these metrics. The metrics, as extracted from the RAGAS documentation, are as follows:

- **Generation**
 - **Faithfulness**: How factually accurate is the generated answer.
 - **Answer Relevancy**: How relevant is the generated answer to the user query.
- **Retrieval**
 - **Context Precision**: The signal-to-noise ratio of the retrieved context.
 - **Context Recall**: Presence of all relevant information needed to answer the question in the retrieved context.

RAG Triad:

Proposes three metrics to evaluate the **query**, **contexts** and **answer** with respect to each other. This is the framework included in **TruLens** and also similar metrics are contained in **DeepEval** (although with subtle differences). These metrics are:

- **Answer Relevance:** Relevance of the generated answer to the user query.
- **Context Relevance:** Relevance of the retrieved context to the user query.
- **Groundedness:** Degree to which the generated answer is grounded in the retrieved context.



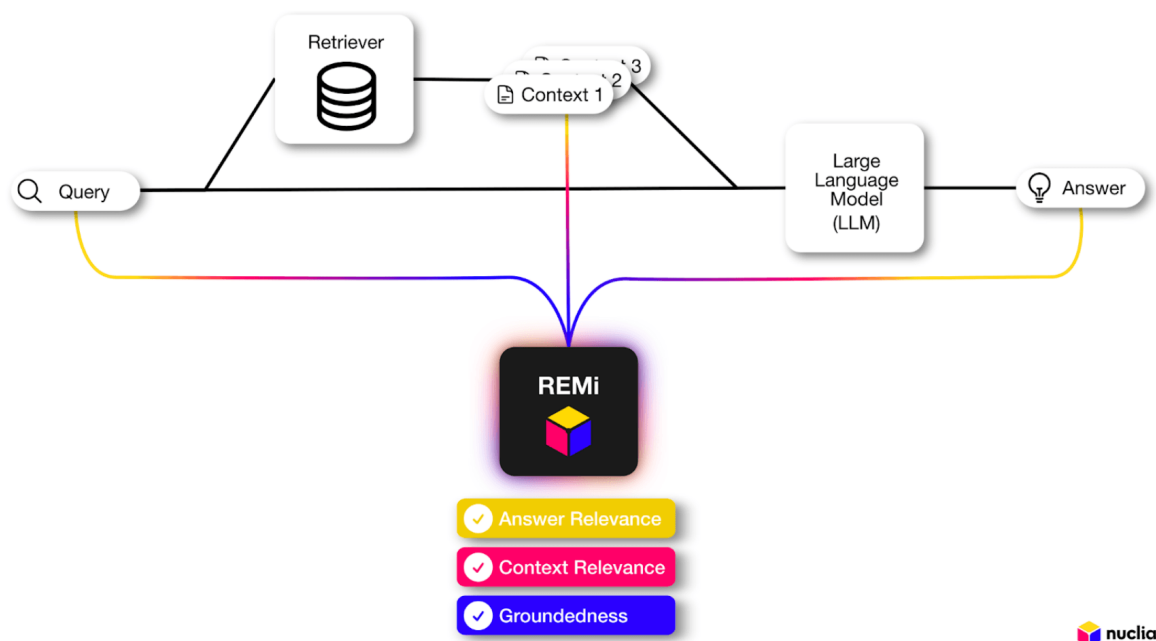
Visual representation of the RAG Triad metrics

Nuclia's RAG evaluation: REMi

At Nuclia, we are committed to providing our clients with cutting-edge metrics that help them assess and recognize the quality of the RAG experience. These metrics also allow us to evaluate and track improvements in the overall experience after each update.

However, it is challenging to deliver these advanced metrics to all clients efficiently. This complexity and efficiency cost is because each assessment requires several interactions with large language models (LLMs), which are typically general-purpose, large and sourced from external providers. To this end, we have developed our own model for RAG evaluation, **REMi**, an small

and efficient LLM fine-tuned specifically for providing RAG evaluation metrics. It gives an evaluation aligned with the RAG Triad, with a 6-point scoring for each metric (from 0 to 5), and an additional reasoning for the Answer Relevance score.



Visual representation of REMi in the RAG pipeline

The RAG Triad evaluation for **REMi** was chosen for several reasons, including:

- The RAG Triad metrics are more **interpretable**, making it easier to detect underlying issues and communicate results to clients or stakeholders.
- The “Faithfulness” RAGAS metric tries to take into account factuality through asking the LLM if content in the answer can be inferred from the given context. In RAG scenarios, where the inner knowledge of the evaluator model might not be aligned with that in the target domain, this approach has the potential to provide misleading results. In contrast, the groundedness metric limits itself to a **more objective** information overlap between the context and the answer.

- The “Answer Relevancy” RAGAS metric involves a reverse-engineering of variants of the original question from the generated answer, and then evaluating through using cosine similarity of embeddings these generated questions with the original query. This approach introduces more complexity as well as another model in the evaluation pipeline. In contrast, the RAG triad approach to answer relevance directly compares the generated answer with the original query.
- The “Context Recall” RAGAS metric relies on the LLM to simultaneously identify the individual statements in the generated answer and determine if each context piece contains the information needed to produce those statements. This method might be problematic when using smaller LLMs, which may lack the capacity to accurately perform both these tasks at the same time. Additionally, there can be issues when a statement contains information from more than one context piece, since its evaluation is binary for each context piece and may not account for combined contributions.

REMi is an adapter on top of Mistral AI’s [Mistral 7B v0.3 Instruct](#). It has been fine-tuned on our proprietary dataset. It takes advantage of function-calling/tool-use capabilities to generate a structured output that is consistent and easily parsable to the schema of the desired metrics. We have made it open and available for the community through **Hugging Face**, the community’s go-to platform for sharing and using Machine Learning models.

We have also developed an open-source Python library, [nuclia-eval](#), which facilitates evaluating RAG experiences using REMi. This library is available on [GitHub](#) and Python Package Index ([PyPI](#)). Users can start evaluating their RAG pipelines with our model in a few lines of code, following the [examples](#).

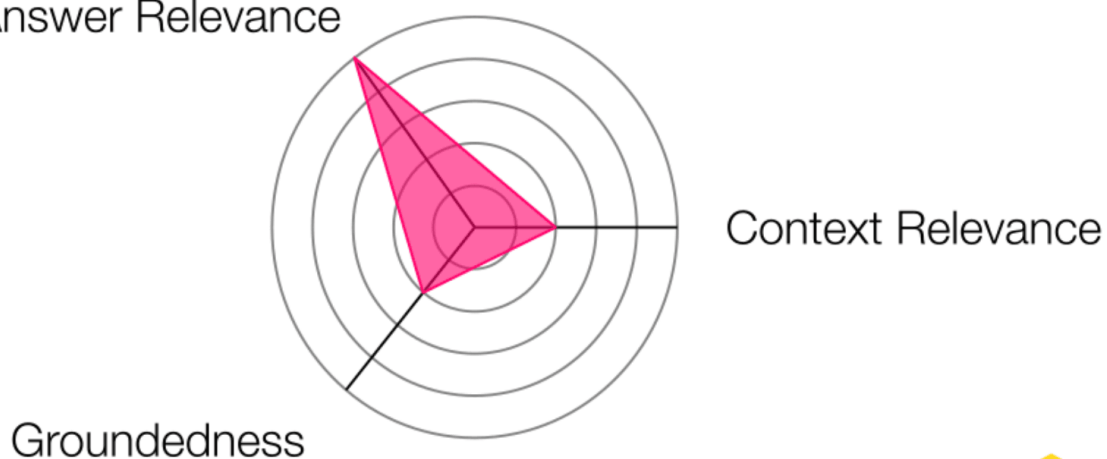
Symptoms of a failing RAG pipeline and solutions

Thanks to the metrics provided by **REMi** and **nuclia-eval**, we can identify the symptoms of a failing RAG pipeline at scale and propose solutions to these issues. We will discuss some of the simpler cases through their radar plots as a starting point. **Nuclia** clients can rely on our expertise to track and troubleshoot the performance of their RAG pipelines and any complex issues that may arise.

The following plots are simplified and exaggerated for illustrative purposes. As a clarification, for each query we would have a single Answer Relevance score and multiple Context Relevance and Groundedness Scores, one for each context piece at play. For the metrics to be observed with a single value for each, an aggregation must be performed along the context pieces for a single query and then across the multiple queries. An example aggregation could be the maximum score for each of the context pieces in a query, and then the average along the queries.

Case 1: Unverifiable Claims

Answer Relevance



Radar chart showing high Answer Relevance but low Context Relevance and Groundedness

Example:

- Query: Which is the best **bakery** in Lyon?

- Context: *Lyon is recognized for its cuisine and gastronomy, as well as historical and architectural landmarks.*
- Generated Answer: *One of the most renowned **bakeries** in Lyon is Pralus, famous for its Praluline.*

The generated answer answers the query but is not grounded in the context piece, which does not contain information about bakeries in Lyon.

Diagnosis: The retrieval model is not able to find the relevant information to answer the query. Despite this, the LLM is able to generate a relevant answer, possibly due to its ability to use its inner knowledge. This is a dangerous situation, as there is no way to verify the information in the generated answer. The model might be hallucinating or generating information that is not necessarily aligned with the domain-specific knowledge that the pipeline is built for. This type of issue could go unnoticed if only the Answer Relevance metric is used, or if the user does not have the domain knowledge to verify the generated answer.

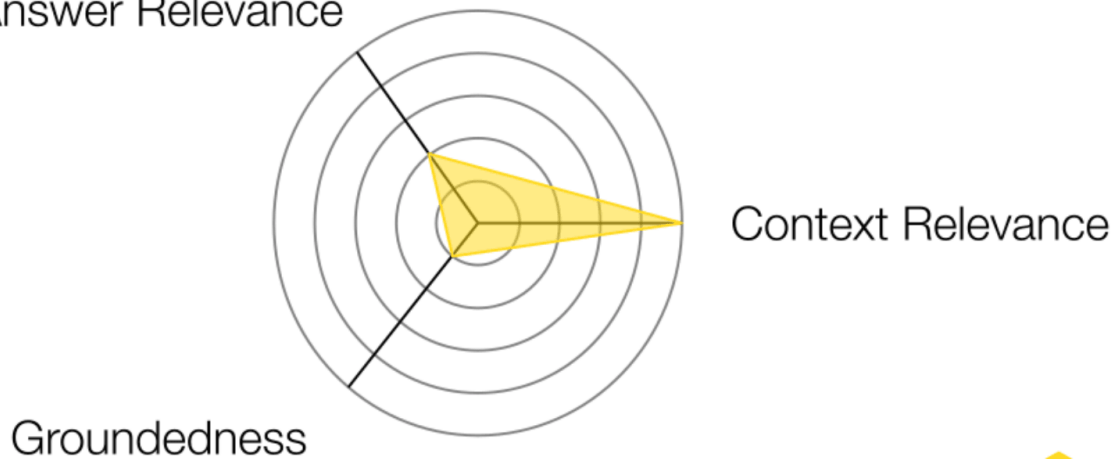
Solution: First, **verify that all the necessary information is present** in the knowledge database. If the information is present, focus on improving retrieval, which can be done in several ways:

1. **Changing the RAG strategy.** For example, at Nuclia, users can choose to retrieve whole documents instead of paragraphs, add visual information by including images as part of the context pieces, etc
2. **Performing data augmentation** on the knowledge base by enriching it with more information. This can be done by adding document summaries, additional metadata to each document, extracting table content, captioning images, among others. All of these features are available in **Nuclia** out-of-the-box.
3. **Improving the retrieval components.** For instance, we by default combine several indexes, such as a full-text search and a semantic search, to retrieve the context pieces. We can rely more heavily on one of them or add new indices with other semantic models. This

new semantic model could be fine-tuned for the domain for even better results.

Case 2: Evasive Response

Answer Relevance



Radar chart showing low Answer Relevance and Groundedness, and high Context Relevance

Example:

- Query: *Which are the two official languages of Eswatini?*
- Context: *Eswatini was a British protectorate until its independence in 1968. They maintain the official language established during that period by their colonizers, together with Swati, their native language.*
- Generated Answer: *The context does not give enough information about the topic to answer the question.*

The context piece contains the information needed to answer the query, but requires a small leap in logic to connect the dots. The LLM is either not able or not willing to make this leap, resulting in an evasive response.

Diagnosis: The retrieval model is able to find the relevant information to answer the query, but the generated answer is not relevant to the query. As a consequence, the generated answer is not grounded in the context. This

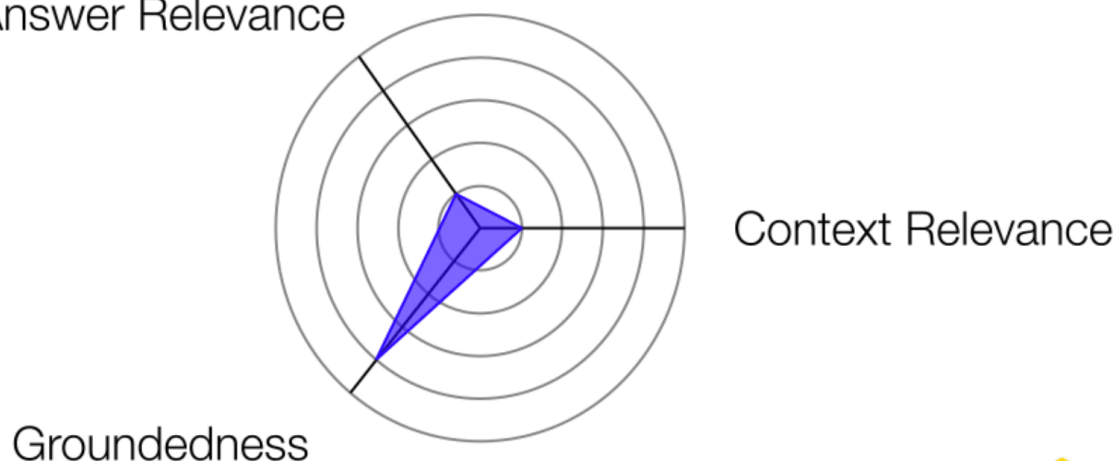
indicates that there might be an issue with LLM, which, even with the relevant information, is not able to generate a relevant answer.

Solution:

1. If we are using a small or efficiency-focused LLM, we should consider **switching to a more powerful LLM** that has better capabilities for piecing together the answer from different chunks of context.
2. **Re-check the system or user prompt templates**, as they might be too restrictive for the LLM to generate a relevant answer. Prompts that include variations of “*Only use the information in the context pieces to generate the answer*” can be helpful to reduce the LLM’s reliance on its inner knowledge, but they can require tuning to avoid limiting the capabilities of the LLM.
3. The context pieces may clash with the LLM’s filters, safeguards, or preference optimization, which can be the case when working in domains such as medicine, criminal law, etc., that may touch on sensitive topics that may be deemed offensive. In these cases, consider **tweaking the safety settings** (if the LLM allows it), using an alternative model that is more permissive, or switching to a domain-specific model.

Case 3: Unrelated Answers

Answer Relevance



Radar chart showing high Groundedness but low Answer and Context Relevance

Diagnosis: The model has generated irrelevant answers to the query, but these answers are grounded in the context. The low context relevance indicates that the context in which the answers are grounded is not relevant to the user's query, explaining the irrelevant answers. This can happen if the wrong context pieces are retrieved, but the LLM still feels compelled to generate an answer based on the available information, disregarding the nuances of the query. This situation is dangerous as it can provide grounded but misleading information to the user.

Example:

- Query: *What is the best **cafe** in Amsterdam?*
- Context: *Leading the charts of the best **coffee shops** in Amsterdam is the Green House.*
- Generated Answer: *The Green House **coffee shop** is a must-visit in Amsterdam.*

The generated answer is grounded in the context piece, but the context piece is not relevant to the user's query, which is about cafes and not coffee shops (which are two very different things in Amsterdam).

Solution: The issue in this case is two-fold. On one hand, the retrieval model failed to find the relevant information. To fix this **refer to the solutions proposed in Case 1** such as changing the RAG strategy, performing data augmentation or improving the retrieval components. On the other hand, the LLM is generating an answer based on the context pieces but not taking the query into account. There are different ways to solve this:

1. **Engineer the prompt template** to indicate to the LLM that it must assess the context pieces in relation to the query before generating the answer. Adding text such as *"Only use the information in the*

context pieces that are relevant to the query to generate the answer” can be helpful.

2. **Ensure that the LLM isn't forced to generate an answer** if it doesn't have enough information to do so, this can be achieved by adding a text to the prompt like *“If the information in the context pieces is not relevant to the query, do not generate an answer”*.
3. **Consider separating context pieces and the query more clearly** in the template, as the LLM might not be differentiating between them properly or might be giving them equal weight. Using separators like Markdown tags or code blocks can help provide a more structured input to the LLM.
4. **Implement the general improvements for the LLM** proposed in Case 2, such as switching to a more powerful model.

Conclusion

In conclusion, Nuclia's introduction of REMi and the nuclia-eval library marks a significant advancement in the evaluation of Retrieval Augmented Generation (RAG) systems. By providing a fine-tuned, efficient open-source LLM specifically for RAG evaluation, REMi addresses the complexities inherent in assessing the outputs of large language models. By following the RAG Triad approach, REMi ensures an interpretable and reliable evaluation process, while the release of nuclia-eval makes these tools accessible to anyone. As the RAG landscape continues to evolve, Nuclia's contributions pave the way for more accurate and efficient evaluation methods, fostering greater trust and effectiveness in these systems.