Is your RAG application performing as well as you expected? Learn to test it with Ragas and PyTest.

Abstract

RAG (Retrieval-Augmented Generation) is a modern technique that enhances response generation by leveraging large language models (LLMs) with data retrieved from an authorized knowledge base. However, evaluating these responses can be challenging because a response may be valid even if it uses different wording each time, or it may simply be a hallucination. This study presents an approach to evaluating both the context used to generate the answers and the answers themselves using a tool called RAGAS. As a result of this research, I was able to build a local RAG system, consume a single data source, generate answers, and produce evaluation metrics using RAGAS and Pytest to assess various quality aspects of the system, such as context precision, context recall, response faithfulness, and response relevance. This research serves as a starting point for further exploration of how to evaluate RAG responses and agents in the future, which is the next objective.

11 Introduction to Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is the process of optimizing the output of a large language model, so it references an authoritative knowledge base outside of its training data sources before generating a response. Large Language Models (LLMs) are trained on vast volumes of data and use billions of parameters to generate original output for tasks like answering questions, translating languages, and completing sentences. RAG extends the already-powerful capabilities of LLMs to specific domains or an organization's internal knowledge base, all without the need to retrain the model. It is a cost-effective approach to improving LLM output so it remains relevant, accurate, and useful in various contexts.

RAG is a useful technique because it leverages the existing capabilities of large LLMs, making it a cost-effective implementation. It also enhances user trust by using reliable data sources and provides greater control over the generated answers.

In simple terms, RAG transforms a dataset into chunks of information. Each chunk is stored in a vector database as embeddings. When a query is made, the system retrieves the most relevant chunks to answer the question, and a response is generated based on the retrieved information.

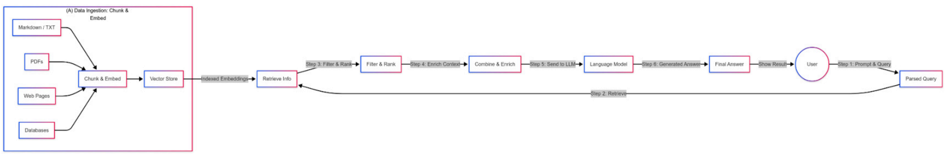

Fig. 1: Diagram explaining how a regular RAG application works

12 Challenges of testing RAG applications

The RAG process uses LLMs to generate responses from a valid knowledge base, but since it still relies on models, the risk of unexpected outputs remains. Additionally, the way users structure their questions can vary significantly from one person to another, making it virtually impossible to test all possible queries

One of the challenges is when the question does not return the required content, and it may be caused when the query is not well-formulated or phrased.

RAG generates responses based on relevant chunks of information using embeddings. However, this can lead to another challenge when some relevant information is not retrieved, resulting in essential content missing from the final response. This may occur if not all necessary chunks of information are retrieved to construct the best response, if the embeddings are not large enough, or if there is an issue with the vector search mechanism itself.

There may also be cases where information is repeated across multiple chunks, and RAG might struggle with retrieving redundant data. This can lead to the model generating responses with duplicated content incoherently.

A RAG app may generate answers based on a vague context. This could be caused by incorrect chunking strategies or the use of poor embedding models.

LLMs have a common problem called hallucinations, which RAG apps can inherit. A hallucination refers to text generation that is unsupported by the retrieved information. In some cases, the generative model may produce text that seems plausible but is not present in the retrieved context. This poses a risk that must be managed.

RAG can produce generic responses despite the good quality of the retrieved chunks of information. This can result in irrelevant text or text that does not fully address the user's needs.

This paper highlights only a few of the challenges, but many more exist. Identifying these challenges and developing effective strategies to manage them is crucial for improving the reliability of RAG applications.

13 Test RAG responses with Ragas and Pytest

RAGAS is a Python library used to generate scores based on different factors by analyzing all components of a RAG application. To generate a score, it uses various parameters depending on the aspect being evaluated. Some of these parameters include the question, the answers, the chunks of information used to generate the response (referred to as the retrieved documents), and, for certain metrics, a ground truth or reference is required.

PyTest is another Python library used as a test framework to write small, readable tests, and it can scale to support complex functional testing for applications and libraries.

3.11 RAGAS & metrics

A metric is a quantitative measure used to evaluate the performance of an AI application. Metrics help in assessing how well the application and individual components that makes up application is performing relative to the given test data. They provide a numerical basis for comparison, optimization, and decision-making throughout the application development and deployment process.

Metrics help identify which part of the application is causing errors or suboptimal performance, making it easier to debug and refine. Metrics enable the tracking of an AI application's performance over time, helping to detect and respond to issues such as data drift, model degradation, or changing user requirements.

Metrics to be reviewed:

1. Context precision
2. Context recall
3. Response faithfulness
4. Response relevancy

3.12 Context precision

Context Precision is a metric that measures the proportion of relevant chunks in the retrieved contexts or relevant chunks of information returned based on the question.

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K} (\text{Precision@k} \times v_k)}{\text{Total number of relevant items in the top } K \text{ results}}$$

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

Fig. 2: Formula to calculate the context precision score

3.13 Context recall

Context Recall measures how many of the relevant documents (or pieces of information) were successfully retrieved. It focuses on not missing important results. Higher recall means fewer relevant documents were left out. In short, recall is about not missing anything important. Since it is about not missing anything, calculating context recall always requires a reference to compare against.

$$\text{Context Recall} = \frac{\text{Number of claims in the reference supported by the retrieved context}}{\text{Total number of claims in the reference}}$$

Fig. 3: Formula to calculate the context recall score

3.14 Response Faithfulness

The Faithfulness metric measures how factually consistent a response is with the retrieved context. It ranges from 0 to 1, with higher scores indicating better consistency. A response is considered faithful if all its claims can be supported by the retrieved context.

$$\text{Faithfulness Score} = \frac{\text{Number of claims in the response supported by the retrieved conte}}{\text{Total number of claims in the response}}$$

Fig. 4: Formula to calculate the response faithfulness

3.15 Response Relevancy

The Response Relevancy metric measures how relevant a response is to the user input. Higher scores indicate better alignment with the user input, while lower scores are given if the response is incomplete or includes redundant information.

$$\text{Faithfulness Score} = \frac{\text{Number of claims in the response supported by the retrieved conte}}{\text{Total number of claims in the response}}$$

Fig. 5: Formula to calculate the response faithfulness

14 How to set up a local RAG system?

This research involved the development of a local RAG application, and the list of requirements is as follows:

```
pypdf
langchain
chromadb # Vector storage
pytest
pytest-asyncio
langchain-openai
langchain-community
langchain-chroma
python-dotenv
ragas
```

Fig. 6: List of requirements to create a local RAG using Python

Embeddings must be obtained from a provider. It is important to consider that a larger embedding model is recommended to improve response generation. For this research, the chosen provider was OpenAI.

```
#Used to get the embedding function.
from langchain_openai import OpenAIEmbeddings


def get_embedding_function():
    embeddings = OpenAIEmbeddings(model="text-embedding-3
large",openai_api_key=openai_api_key)
    return embeddings
```

Fig. 7: Function to use the embeddings from OpenAI and transform data to embeddings

The next step is to ingest data from a data source. For this research, PDF files will be stored in the 'data' folder. The database used to store the embeddings extracted from the PDFs is ChromaDB, a vector database optimized for this type of task. Each PDF and its content will be divided into multiple chunks, which will be saved in the database.

```python
def main():

    # Check if the database should be cleared (using the --clear flag).
    parser = argparse.ArgumentParser()
    parser.add_argument("--reset", action="store_true", help="Reset the database.")
    args = parser.parse_args()
    if args.reset:
        print("✨ Clearing Database")
        clear_database()

    # Create (or update) the data store.
    documents = load_documents()
    chunks = split_documents(documents)
    add_to_chroma(chunks)


#Used to load documents from a directory.
def load_documents():
    document_loader = PyPDFDirectoryLoader(DATA_PATH)
    return document_loader.load()


def split_documents(documents: list[Document]):
    text_splitter = RecursiveCharacterTextSplitter(
        chunk_size=800,
        chunk_overlap=80,
        length_function=len,
        is_separator_regex=False,
    )
    return text_splitter.split_documents(documents)



#Used to add documents to the Chroma database.
def add_to_chroma(chunks: list[Document]):
    # Load the existing database.
    db = Chroma(
        persist_directory=CHROMA_PATH, embedding_function=get_embedding_function()
    )

    # Calculate Page IDs.
    chunks_with_ids = calculate_chunk_ids(chunks)

    # Add or Update the documents.
    existing_items = db.get(include=[])  # IDs are always included by default
    existing_ids = set(existing_items["ids"])
    print(f"Number of existing documents in DB: {len(existing_ids)}")

    # Only add documents that don't exist in the DB.
    new_chunks = []
    for chunk in chunks_with_ids:
        if chunk.metadata["id"] not in existing_ids:
            new_chunks.append(chunk)

    if len(new_chunks):
        print(f"👉 Adding new documents: {len(new_chunks)}")
        new_chunk_ids = [chunk.metadata["id"] for chunk in new_chunks]
        db.add_documents(new_chunks, ids=new_chunk_ids)
        db.persist()
    else:
        print("✅ No new documents to add")

# Used to calculate the chunk IDs.
def calculate_chunk_ids(chunks):

    # This will create IDs like "data/monopoly.pdf:6:2"
    # Page Source : Page Number : Chunk Index
    # - Page Source: The source of the page.


    last_page_id = None
    current_chunk_index = 0

    for chunk in chunks:
        source = chunk.metadata.get("source")
        page = chunk.metadata.get("page")
        current_page_id = f"{source}:{page}"

        # If the page ID is the same as the last one, increment the index.
        if current_page_id == last_page_id:
            current_chunk_index += 1
        else:
            current_chunk_index = 0

        # Calculate the chunk ID.
        chunk_id = f"{current_page_id}:{current_chunk_index}"
        last_page_id = current_page_id

        # Add it to the page meta-data.
        chunk.metadata["id"] = chunk_id

    return chunks

#Used to clear the database.
def clear_database():
    if os.path.exists(CHROMA_PATH):
        shutil.rmtree(CHROMA_PATH)


if __name__ == "__main__":
    main()
```

Fig. 8: Functions in charge of ingesting PDF files, splitting the documents into chunks, and then saving the embeddings in the database.

Once the embeddings are in the database, it is time to ask questions and get responses from the RAG system.

The RAG system receives the query, retrieves the embeddings from the database, and then sends the question along with the context to a model such as GPT-4o-mini. It then generates an answer based on the most relevant chunks of information retrieved from the database.

```python
# The prompt template for the chat openAI model.
PROMPT_TEMPLATE = """
Answer the question based only on the following context:

{context}

---

Answer the question based on the above context: {question}
"""

def main():
    # Create CLI to get the query text. For example: python query.py "What is the
capital of France?"
    parser = argparse.ArgumentParser()
    parser.add_argument("query_text", type=str, help="The query text.")
    args = parser.parse_args()
    query_text = args.query_text
    query_rag(query_text)

def query_rag(query_text: str):
    embedding_function = get_embedding_function()
    db = Chroma(persist_directory=CHROMA_PATH,
embedding_function=embedding_function)

    results = db.similarity_search_with_score(query_text, k=5)

    # - Join the page content of the results into a single string.
    context_text = "\n\n---\n\n".join([doc.page_content for doc, _ in results])

    # - Create a chat prompt template.
    prompt_template = ChatPromptTemplate.from_template(PROMPT_TEMPLATE)
    # - Send the context and question to the model.
    messages = prompt_template.format_messages(context=context_text,
question=query_text)

    # Instantiate the ChatOpenAI model.
    model = ChatOpenAI(
        # - Temperature is the randomness of the model. 0 is deterministic, 1 is
random.
        temperature=0,
        # - Model name.
        model_name="gpt-4o-mini",
        # If not set in your environment, you can hardcode your API key here:
        openai_api_key=os.environ.get("OPENAI_API_KEY")
    )


    response = model.invoke(messages)

    response_text = response.content.strip()

    # Build a list of retrieved documents in the desired JSON structure
    retrieved_docs = []
    for doc, _ in results:
        file_name = doc.metadata.get("id", "Unknown File Name")
        retrieved_docs.append({
            "file_name": file_name,
            "page_content": doc.page_content
        })


    # Build the final JSON response
    output = {
        "answer": response_text,
        "retrieved_docs": retrieved_docs
    }

    json_output = json.dumps(output, indent=2, ensure_ascii=False)
    return json_output

if __name__ == "__main__":
    main()
```

Fig. 9: Function to answer a question using the RAG application using the embeddings, with the power of an LLM such as GPT-4o-mini.

15 How to set up Ragas and Pytest?

RAGAS requires different parameters depending on the evaluated metric to generate a score. The majority of metrics will require the following parameters:

1. Question
2. Response
3. Retrieved contexts

4. Reference

```python
@pytest.mark.asyncio
async def test_context_precision(langchain_llm_ragas_wrapper, get_question, print_log):

    # Get Question
    question = get_question("context_precision", "simple")

    # Get Response
    response = query_rag(question)
    parsed_response = json.loads(response)

    # Initialize the LLM and Ragas Setup for Context Precision
    context_precision = LLMContextPrecisionWithoutReference(llm=langchain_llm_ragas_wrapper)

    # Feed Data
    sample = SingleTurnSample(
        user_input=question,
        response=parsed_response["answer"],
        retrieved_contexts= [doc["page_content"] for doc in parsed_response["retrieved_docs"]],
    )

    # Score
    score = await context_precision.single_turn_ascore(sample)
    print_log(question, parsed_response["answer"], parsed_response["retrieved_docs"], score=score)
    assert score >= 0.5
```

Fig. 10: Context precision test structure with its assertion

```python
@pytest.mark.asyncio
async def test_context_recall(langchain_llm_ragas_wrapper, get_question, get_reference, print_log):

    # Get Question
    question = get_question("context_recall", "simple")

    # Get Reference
    reference = get_reference("context_recall", "simple_reference")

    # Get Response
    response = query_rag(question)
    parsed_response = json.loads(response)

    # Initialize the LLM and Ragas Setup for Context Precision
    context_recall = LLMContextRecall(llm=langchain_llm_ragas_wrapper)

    # Feed Data
    sample = SingleTurnSample(
        user_input=question,
        retrieved_contexts= [doc["page_content"] for doc in parsed_response["retrieved_docs"]],
        reference=reference
    )

    # Score
    score = await context_recall.single_turn_ascore(sample)
    print_log(question, parsed_response["answer"], parsed_response["retrieved_docs"], reference, score)
    assert score >= 0.5
```

Fig. 11: Context recall test structure with its assertion. Notice how it requires a reference to validate the relevance of the context retrieved.

```python
@pytest.mark.asyncio
async def test_faithfulness(langchain_llm_ragas_wrapper, get_question, print_log):

    # Get Question
    question = get_question("faithfulness", "simple")

    # Get Response
    response = query_rag(question)
    parsed_response = json.loads(response)

    # Initialize the LLM and Ragas Setup for Context Precision
    faithfulness = Faithfulness(llm=langchain_llm_ragas_wrapper)

    # Feed Data
    sample = SingleTurnSample(
        user_input=question,
        response=parsed_response["answer"],
        retrieved_contexts= [doc["page_content"] for doc in parsed_response["retrieved_docs"]],
    )

    # Score
    score = await faithfulness.single_turn_ascore(sample)
    print_log(question, parsed_response["answer"], parsed_response["retrieved_docs"], score=score)
    assert score >= 0.5
```

Fig. 12: Response faithfulness test structure with its assertion.

```python
@pytest.mark.asyncio
async def test_response_relevancy(langchain_llm_ragas_wrapper, get_embeddings, get_question,
print_log):

    # Get Question
    question = get_question("response_relevancy", "simple")

    # Get Response
    response = query_rag(question)
    parsed_response = json.loads(response)

    # Initialize the langchain wrapper and embeddings to be used for the response relevancy metric
    response_relevancy = ResponseRelevancy(llm=langchain_llm_ragas_wrapper, embeddings=get_embeddings)


    # Feed Data
    sample = SingleTurnSample(
        user_input=question,
        response=parsed_response["answer"],
        retrieved_contexts= [doc["page_content"] for doc in parsed_response["retrieved_docs"]],
    )

    # Score
    score = await response_relevancy.single_turn_ascore(sample)
    print_log(question, parsed_response["answer"], parsed_response["retrieved_docs"], score=score)
    assert score >= 0.5
```

Fig. 13: Response relevancy test structure with its assertion.

Each test receives different parameters to process the score, such as a LangChain wrapper, the embeddings to measure the quality of the response, an external JSON file where the questions are centralized, and a print log function to track what is happening behind the scenes.

16 Analyzing the scores and interpreting the results

For this project, I decided to use a simple dataset—a PDF file containing various information about cats. That's why you will see questions related to this topic. However, we can ingest any PDF for testing purposes.

Let's go through the different metrics, ask a simple question for each, check the score, and figure out why it got that result.

6.11 Context precision

For context precision, I asked the following question:

```
#Question: What physical traits of domestic cats are unique compared to wild
felines?
```

Fig. 14: Context precision question for demonstration.

The response generated was:

```
# Answer: Domestic cats exhibit several physical traits that are unique compared to
wild felines, including:
    # 1. **Coloration**: Domestic cats have a wider range of colors and patterns
due to less need for camouflage in captivity compared to their wild counterparts.
    # 2. **Size**: Domestic cats tend to be smaller in size than many wild felines,
which is attributed to changes in diet and habitat.
    # 3. **Brain Size**: Domestic cats have a smaller brain size due to the gradual
elimination of unnecessary survival instincts that are more critical in the wild.
    # 4. **Physical Characteristics**: There is a greater variation in body shape
and size among domestic cats, influenced by differing gene pools across countries.
```

Fig. 15: Response generated from the previous question

As explained earlier, this answer is based on relevant chunks of information retrieved by the RAG application. Take a look at the relevant chunks of information:

```
# Retrieved Contexts:

    # File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:63:3
    # Content: protection from humans, in the sense that they would be safe from
other predators as long
    # as they remained near human habitats. These two species eventually fused to
create a new
    # breed of cat, related to the modern-day Egyptian Mau.
    # The change in temperament is attributed to two principal factors: heredity and
learned
    # tolerance of humans. The changes due to domestication follow a pattern similar
to other
    # domesticated animals including wolves (dogs), and cattle. These changes
include coloration
    # as there is less need for camouflage in captivity than in the wild, smaller
brain size due to the
    # gradual elimination of unnecessary survival instincts, and an overall decrease
in size due to
    # the change in diet and habitat.

    # File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:27:1
    # Content: gesturing. Because the domestication of the cat is relatively recent,
cats may also still live
    # effectively in the wild, often forming small colonies. The cat's association
with humans leads
    # it to figure prominently in the mythology and legends of several cultures,
including the
    # ancient Egyptians, Vikings, and Chinese.
    # Characteristics
    # Physical
    # Cats typically weigh between 2.5 and 7 kg (5.5-16 lb); however, some breeds,
such as the
    # Maine Coon can exceed 11.3 kg (25 pounds). Some have been known to reach up to
23 kg
    # (50 lb), due to overfeeding. This is very unhealthy for the cat, and should be
prevented
    # through diet and exercise (playing), especially for cats living exclusively
indoors.
    # In captivity, indoor cats typically live 15 to 20 years, though the oldest-
known cat lived

    # File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:34:2
    # Content: vinyl nail caps that are affixed to the claws with nontoxic glue,
requiring periodic
    # replacement when the cat sheds its claw sheaths (usually every four to six
weeks).
    # Environment
    # The wild cat, ancestor of the domestic cat, is believed to have evolved in a
desert climate,
    # as evident in the behavior common to both the domestic and wild forms. Wild
cats are native
    # to all continents other than Australasia and Antarctica. Their feces are
usually dry, and cats
    # prefer to bury them in sandy places. They are able to remain motionless for
long periods,
    # especially when observing prey and preparing to pounce. In North Africa there
are still small
    # wildcats that are probably related closely to the ancestors of today's
domesticated breeds.

    # File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:246:1
    # Content: can be any colour or combination of colours. They also exhibit a wide
range of physical
    # characteristics, and as a result, domestic shorthaired cats in different
countries tend to look
    # different in body shape and size, as they are working from differing gene
pools. However,
    # they are all recognizable as cats, and any male (tom) cat could successfully
breed with any
    # other female (queen), meaning they are the same species.
    # See also
    # • Cat coat genetics
    # • Domestic longhaired cat
    # Home | Up

    # Farm Cat
    # Farm cats are cats used for catching pests on farms. They are feral cats,
meaning that
    # they are wild and you should have caution around them. It depends on the
farmer if they will
    # be treated well, like with food and water, or just being there to do what they
are supposed to

    # File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:111:1
    # Content: crossbreeding, hybridizing domestic cats with desired coat and
temperament features with
    # Asian Leopard Cats (ALC) and ALC hybrids. The principle of hybrid vigor
dictates that hybrid
    # cats are often larger than either parent, but are typically infertile. F1 and
F2 males are nearly
    # always infertile, F3 males are normally infertile, but females are often
fertile even in early
    # hybrids.
    # A cat with one wild ancestor is called an F1, short for first filial. An F1
bred with a
    # domestic cat or other bengal filial cat yields an F2, or second filial. Any
kittens from an F2
    # female are termed F3. Any kittens from an F3 female are termed F4. F4 and
higher
    # generations are officially known as Stud Book Tradition (SBT) bengals and can
be shown and
```

Fig. 16: Every chunk of information retrieved is used as context to generate the response.

Context precision measures how much of the retrieved context is relevant to answering the question. If the score is high, it means that most of the retrieved contexts are useful for answering the query. If the score is low, it means that many of the retrieved contexts are irrelevant or only partially relevant.

For this question, the context precision score was 0.75(75%), meaning that the retrieved context was highly relevant to the question. However, there's still room for improvement since some retrieved chunks weren’t entirely relevant. For example, one passage discussed coat color diversity in domestic cats, which isn’t related to what makes them unique compared to wild cats. Another passage focused on hybrid breeding with wild cats—interesting, but not directly answering the question.

Also, the response missed some key physical differences. For instance, domestic cats have shorter snouts and smaller teeth compared to wild felines. Their tails tend to be more upright when communicating, unlike wild cats. Their fur is also softer since they don’t have to survive in harsh outdoor conditions.

6.12 Context recall

For context precision, I asked the following question:

```
# Question: What are the different types of cat coats, fur textures, and patterns,
and how do they affect a cat's appearance?
```

Fig. 17: Context recall question for demonstration.

The response generated was:

```
# Answer: The context describes several types of cat coats, fur textures, and
patterns, which significantly affect a cat's appearance:
# 1. **Tabby Cat Patterns**:
#    - **Blotched Tabby**: Features a classic pattern with butterflies and bull's-
eyes, common in NW Europe.
#    - **Mackerel Tabby**: Characterized by vertical stripes down the sides,
resembling a fish.
#    - **Spotted Tabby**: A variation where the stripes are broken into spots.
# 2. **Tortoiseshell and Calico**:
#    - These coats feature three colors mottled throughout, also known as "tortie."
A true tortoiseshell has specific color combinations that create a unique
appearance.
# 3. **Fur Length and Texture**:
#    - Governed by the Long hair gene, where the dominant form (L) results in short
hair and the recessive form (l) leads to long hair.
#    - There are also genes that produce curly-coated or "rex" cats, which have a
unique texture that is not silky but has a drag like velvet. The coat is described
as loose, bouncy, and airy, often standing away from the body.
# 4. **Coat Characteristics**:
#    - The coat can vary by season and maturity, typically being wavy or curly,
with the longest curls found in the ruff and neck area.
#    - Longhairs have a curly plumed tail, while shorthairs have tails resembling
bottle brushes. Both types have long curled whiskers.
# These variations in coat types, patterns, and textures contribute to the overall
aesthetic appeal of cats, influencing how they are perceived and appreciated by cat
lovers.
```

Fig. 18: Response generated from the previous question

As explained earlier, this answer is based on relevant chunks of information retrieved by the RAG application. Take a look at the relevant chunks of information:

```
# Retrieved Contexts:

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:39:1
# Content: large ears and very short sleek fur).
# Tabby cat
# Striped, with a variety of patterns. The classic "blotched tabby" pattern is the
most
# common and consists of butterflies and bull's -eyes. The mackerel tabby is a
series of
# vertical stripes down the cat's side (resembling the fish). This pattern broken
into spots
# is referred to as spotted tabby. The worldwide evolution of the cat means that
certain
# types of tabby are associated with certain countries; for instance, blotched
tabbies are
# quite rare outside NW Europe, where they are the most common type.
# Tortoiseshell and Calico
# Featuring three colors mottled throughout the coat, this cat is also known as a
Calimanco
# cat or Clouded Tiger cat, and by the nickname "tortie". A true tortoiseshell must
consist

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:55:1
# Content: as yet unidentified, believed to result in different degrees of shading,
some more
# desirable than others.
# Genes involved in fur length and texture
# Cat fur length is governed by the Long hair gene in which the dominant form, L
codes for
# short hair, and the recessive l codes for long hair.
# There are many genes resulting in unusual fur. These genes were discovered in
random-
# bred cats and selected for. Some of the genes are in danger of going extinct
because the
# breeders have not marketed their cats effectively, the cats are not sold beyond
the region
# where the mutation originated, or there is simply not enough demand for the
mutation.
# There are various genes producing curly coated or "rex" cats. New types of rex pop
up

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:51:2
# Content: the Wikipedia.
#
# Cat Coat Genetics
#
#  Back | Home | Up | Next
# Home | Up
#
# The genetics of cat coat coloration, pattern, length, and texture is a complex
subject, and
# many different genes are involved.
# Genes involved in albinism, dominant white, and
# white spotting
# • The dominant C gene and its recessive alleles determine whether a cat is a
# complete albino (either pink-eyed or blue-eyed), a temperature sensitive albino
# (Burmese, Siamese, or a blend known as Tonkinese), or a non-albino. If a cat has

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:133:4
# Content: Tabby points are especially attractive. Newer varieties such as ticked
tabbies, shadeds and
# darker points are also being bred. The curl tends to open up the coat showing off
shading,
# ticking or silver undercoats.
# The coat itself is described as having a unique textured feel. It is not silky,
having a certain
# drag on the hand like velvet and the texture comes as much from the shape of the
curls as
# from the mixture of different hair types. It should be soft and inviting, although
the shorthairs

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:134:0
# Content: NICOLAE SFETCU: ABOUT CATS
# 133
# will have more texture to their coats. The coat is rather loose and bouncy often
feeling
# springy when patted, and stands away from the body with no thick undercoat. It is
light and
# airy and judges sometimes blow on the coat to see if it will part. The coat varies
according to
# the season and the maturity of the cat but is essentially wavy or curly all over
with the longest
# and most defined curls in the ruff and on the neck often falling in ringlets.
There are also
# curly ear furnishings including tufts at the ear tips and ear muffs. The longhairs
have a curly
# plumed tail while the shorthairs have tails rather like bottle brushes, and both
have long
# curled whiskers. Sometimes the coat falls into a natural parting along the bac k,
jokingly
```

Fig. 19: Every chunk of information retrieved is used as context to generate the response.

For context recall, a reference or ground truth is required to determine whether the retrieved chunks cover all the necessary details.

```
# Reference(truth): Cats have various coat types, fur textures, and patterns that
influence their appearance. Coat types include short-haired, long-haired, semi-
longhaired, curly (like Rex breeds), and hairless (like the Sphynx). Fur textures
vary from soft and silky (Persian) to coarse (Abyssinian) or wavy (Devon Rex).
Common fur patterns include solid (one color), tabby (striped or swirled), bicolor
(two colors), tortoiseshell (blended black and orange), calico (tortoiseshell with
white), pointed (darker face, ears, paws, and tail, like Siamese), and spotted
(like Bengals). These traits impact a cat's appearance by influencing color
distribution, texture feel, and overall distinctiveness, with genetics playing a
key role in determining variations.
```

Fig. 20: Reference with all the necessary details to answer a question, it is going to help RAGAS to calculate the context recall metric.

Context recall measures how much of the total relevant information was retrieved. For this question, the context recall score was 0.4 (40%), which isn't great.

Analyzing the results, there are several reasons why the score is low. A lot of important details were left out. The system didn't mention all coat types, such as solid-colored cats, bicolor, pointed coats (like Siamese), or spotted coats (like Bengals). It also missed some fur textures, only discussing curly coats like those of Rex breeds but failing to mention silky Persians or coarse Abyssinians. Hairless cats were barely addressed, with the Sphynx being completely left out. Additionally, there was no real discussion on genetics, even though coat colors and patterns are largely genetic, and the system didn't retrieve enough information on this aspect.

It captured part of the answer but not the full picture, which is why the recall score is low.

6.13 Response Faithfulness

To evaluate faithfulness, I asked the following question:

```
# Question: What are some unique physical traits and abilities of cats?
```

Fig. 21: Faithfulness question for demonstration.

The response generated was:

```
# Answer: Some unique physical traits and abilities of cats include:
# 1. **Retractable Claws**: Cats have claws that are sheathed in skin and fur when
relaxed, which helps keep them sharp.
# They can extend their claws involuntarily by stretching or swatting at prey.
# 2. **Superior Night Vision**: Cats possess a tapetum lucidum, which enhances
their ability to see in low light conditions,
# although it may reduce visual acuity in bright light.
# 3. **Acute Sense of Balance**: Cats can reflexively twist their bodies during a
fall to right themselves,
# thanks to their acute sense of balance and flexibility.
# 4. **Digitigrade Locomotion**: Cats walk on their toes, with the bones of their
feet making up the lower part of their visible legs.
# This allows them to walk quietly and precisely, minimizing noise and visible
tracks.
# 5. **Whiskers**: Cats have about a dozen whiskers in four rows on each upper lip,
which are sensitive and help them navigate their environment.
# 6. **Strong Sense of Smell**: A cat's sense of smell is about 14 times stronger
than that of humans,
# aided by having twice as many smell-sensitive cells in their noses.
# 7. **Vomeronasal Organ**: Cats have a specialized scent organ in the roof of
their mouths that enhances their ability to detect scents,
# which they access through a behavior called gaping.
# These traits contribute to their effectiveness as predators and their
adaptability in various environments.
```

Fig. 22: Response generated from the previous question

As explained earlier, this answer is based on relevant chunks of information retrieved by the RAG application. Take a look at the relevant chunks of information:

```
# Retrieved Context:

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:28:3
# Content: they are superior in many ways to those of humans. These along with the
cat's highly
# advanced eyesight, taste, and touch receptors make the cat extremely sensitive
among
# mammals.
# Sight
# Testing indicates that a cat's vision is superior at night in comparison to
humans, and
# inferior in daylight. Cats, like dogs, have a tapetum lucidum that reflects extra
light to the
# retina. While this enhances the ability to see in low light, it appears to reduce
net visual
# acuity, thus detracting when light is abundant. In very bright light, the slit-
like iris closes very
# narrowly over the eye, reducing the amount of light on the sensitive retina, and
improving
# depth of field. The tapetum and other mechanisms give the cat a minimum light
detection

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:28:2
# Content: and visible tracks.
# Like many predators, cats have retractable claws. This is actually a misnomer
because in
# their normal, relaxed position the claws are sheathed with the skin and fur around
the toe
# pads. This is done to keep the claws sharp by preventing wear from contact with
the ground.
# It is only by stretching, such as swatting at prey, that the connecting tendons
are pulled taut,
# forcing the claws to extend. Thus extending the claws is an involuntary action.
# Senses
# Measuring the senses of any animal can be difficult, because there is usually no
explicit
# communication (e.g., reading aloud the letters of a Snellen chart) between the
subject and the
# tester.
# While a cat's senses of smell and hearing may not be as keen as, say, those of a
mouse,

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:29:2
# Content: inches (7.5 cm) the location of a sound being made one yard
(approximately one meter)
# away.
# Smell
# A domestic cat's sense of smell is about 14 times stronger than a human's. Cats
have twice
# as many smell-sensitive cells in their noses as people do, which means they can
smell things
# we are not even aware of. Cats also have a scent organ in the roof of their mouths
called the
# vomeronasal, or Jacobson's, organ. When a cat wrinkles its muzzle, lowers its
chin, and lets
# its tongue hang a bit, it is opening the passage to the vomeronasal. This is
called gaping.
# Gaping is the equivalent of the Flehmen response in other animals, such as dogs
and horses.
# Touch
# Cats generally have about a dozen whiskers in four rows on each upper lip, a few
on each

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:27:1
# Content: gesturing. Because the domestication of the cat is relatively recent,
cats may also still live
# effectively in the wild, often forming small colonies. The cat's association with
humans leads
# it to figure prominently in the mythology and legends of several cultures,
including the
# ancient Egyptians, Vikings, and Chinese.
# Characteristics
# Physical
# Cats typically weigh between 2.5 and 7 kg (5.5-16 lb); however, some breeds, such
as the
# Maine Coon can exceed 11.3 kg (25 pounds). Some have been known to reach up to 23
kg
# (50 lb), due to overfeeding. This is very unhealthy for the cat, and should be
prevented
# through diet and exercise (playing), especially for cats living exclusively
indoors.
# In captivity, indoor cats typically live 15 to 20 years, though the oldest-known
cat lived

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:28:1
# Content: A popular belief holds that cats always land on their feet. They do
usually, but not always.
# During a fall, a cat can reflexively twist its body and right itself using its
acute sense of balance
# and flexibility. [8] It always rights itself in the same way, provided it has the
time to do so
# during a fall. Certain breeds that don't have a tail are a notable exception,
since a cat moves
# its tail and relies on conservation of angular momentum to set up for landing.
# Cats, like dogs, are digitigrades: they walk directly on their toes, the bones of
their feet
# making up the lower part of the visible leg. They are capable of walking very
precisely,
# placing each hind paw directly in the print of the corresponding forepaw,
minimising noise
# and visible tracks.
```

Fig. 23: Every chunk of information retrieved is used as context to generate the response.

Faithfulness measures whether the response introduces external or incorrect information. It ensures that the response only rephrases and summarizes what is found in the retrieved context. A high score means the response is accurate and directly supported by the retrieved context. A low score indicates that the response includes information not supported by the retrieved context, potentially altering key facts and leading to a misleading or incorrect answer.

For this question, the faithfulness score was 0.86 (86%), meaning that most of the response is accurate and well supported by the retrieved context. However, there are some areas for improvement, such as slight rewording or extrapolation beyond the retrieved text and missed opportunities to expand on certain details.

6.14 Response relevancy

To evaluate response relevancy, I asked the following question:

Fig. 24: Response relevancy question for demonstration.

The response generated was:

```
# Answer:
# Cats communicate with humans and other cats through a variety of vocalizations,
body language, and scent marking.

# 1. **Vocalizations**: Cats can produce about 100 different sounds, including
meows, purrs, and other calls.
# They use these sounds to indicate pain, request attention (such as being fed or
played with), or as a greeting.
# The pitch and tone of their calls can change as they age, starting from high-
pitched squeaks in kittens to deeper sounds in adults.
# 2. **Body Language**: Cats use non-verbal cues to communicate their feelings and
needs.
# For example, whiskers pointing forward indicate curiosity and friendliness, while
whiskers lying flat suggest defensiveness or aggression.
# Other behaviors include rubbing against humans for affection, making eye contact
to signal a need, and performing tricks similar to dogs.
# 3. **Scent Marking**: Cats have scent glands around their mouths and other areas,
which they use to mark their humans as part of their territory.
# This behavior is a way of claiming their human and establishing a bond.
# Overall, cats exhibit a combination of vocal and non-verbal communication methods
to express their needs and emotions to both humans and other cats.
```

Fig. 25: Response generated from the previous question

As explained earlier, this answer is based on relevant chunks of information retrieved by the RAG application. Take a look at the relevant chunks of information:

```
# Retrieved Context:

# Retrieved Contexts:
# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:36:2
# Content: a trip to the litter box before bedtime and snuggling up close to its
companion in bed or on
# the sofa. Oth er behaviors could include mimicking sounds of the owner or using
certain
# sounds the cat picks up from the human; sounds representing specific needs of the
cat, which
# the owner would recognize. The cat may also be capable of learning to communicate
with
# the human using non -spoken language or body language  such as rubbing for
affection
# (confirmation), facial expressions and making eye -contact with the owner if
something
# needs to be addressed (e.g . finding a bug crawling on the floor for the owner to
get rid of).
# Some owners like to train their cat to perform "tricks" commonly exhibited by dogs
such as
# jumping.

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:36:1
# Content: communications skills are required of the lone hunter. Thus,
communicating with s uch an
# animal is problematic, and cats in particular are labelled as opaque or
inscrutable, if not
# obtuse, as well as aloof and self -sufficient. However, cats can be very
affectionate towards
# their humans, especially if they imprint on them at a very young  age and are
treated with
# consistent affection.
# Human attitudes toward cats vary widely. Some humans keep cats for companionship
as
# pets. Some people (known as cat lovers) go to great lengths to pamper their cats,
sometimes
# treating them almost as if they were children. When a cat bonds with its human
owner, at
# times, the cat may display behaviors similar to that of the human. Such behavior
may include

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:30:2
# Content: depending on meaning. Usually cats call out to indicate pain, request
human attention (to be
# fed or played with, for example), or as a greeting. Some cats are very vocal, and
others rarely
# call out. Cats are capable of about 100 different vocalisations, compared to about
10 for dogs.
# A kitten's call first starts out as a high -pitched squeak-like sound when very
young, and
# then deepens over time. Some cats, h owever, do not exercise their voices a lot,
so their call
# may remain similar to that of a kitten through adulthood.
# Cats can also produce a purring noise that typically indicates that the cat is
happy, but
# also can mean that it feels distress. Cats purr among other cats—for example, when
a mother

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:30:0
# Content: NICOLAE SFETCU: ABOUT CATS
# 29
# Whiskers are also an indication of the cat's attitude. Whiskers point forward when
the
# cat is inquisitive and friendly, and lie flat on th e face when the cat is being
defensive or
# aggressive.
# Taste
# According to National Geographic (December 8), cats cannot taste sugary foods due
to a
# faulty sweet receptor gene. Some scientists believe this is related to the cat's
diet being
# naturally high in protein, though it is unclear whether it is the cause or the
result of it.
# Communication
# The unique sound a small cat makes is written onomatopoeically as "meow" in
American
# English; "meow" or "miaow" in British English; "miaou" or "miaw" in French; "miao
" in
# Mandarin Chinese and Italian; "miau" in German, Spanish, Finnish, Lithuanian,
Polish,

# File: data/About_Cats-Nicolae_Sfetcu-CCNS.pdf:93:2
# Content: food. It is also a way of 'marking' its human as its very own. Using
scent glands located around
# its mouth and elswhere, it subtly 'marks' its human as part of its cat territory.
Most cats
# prefer gentle rubs behind the ears. To inform their humans they need petting or
attention, a
# cat may push its entire body weight up against the human as the cat snuggles next
to his/her
# favorite person.
# Some subtle Anthropomorphisms
# • Disgust - Lifting and subsequent shaking of a paw or paws is sign of disgust.
The
# more paws the more disgusting. This can sometimes be a four paw affair wit h
# each paw being lifted and shaken before the other.
# • Agitation - The swishing or sweeping of the tail in one full 180 degree swoop
mid-
```

Fig. 26: Every chunk of information retrieved is used as context to generate the response.

Response relevancy measures how well the generated response answers the user's question based on the retrieved context. If the score is high, it means that the response fully answers the user's question with clear, direct, and specific information.

For this question, the context precision score was 0.99 (99%), meaning that the response aligns perfectly with the retrieved context. There are no fabrications or overgeneralizations, and it maintains good structure and readability.

17 Case study and conclusions

This research was applied to a simple dataset, but it serves as a good starting point for evaluating the scores and determining whether they make sense after analysis. Additionally, by using simple metrics, we validate different aspects of a response generated by an LLM without explicitly specifying a reference—except for context recall, where it is necessary.

After the scores are generated, it may be necessary to adjust various aspects to achieve higher scores. Possible solutions include refining the question using prompt techniques (depending on who will be using the RAG app—it is essential to understand the final user), making improvements to the RAG app itself, increasing the number of relevant information chunks, using a larger embedding model, or optimizing the ranking system that retrieves the chunks for better efficiency.

There is room for improvement in this research. A good next step could be to increase the number of data sources and explore the possibility of testing a conversation between a human and AI, as RAGAS can evaluate it. That will be the next step.

References

[RAG 24] RAGAS Documentation: Metrics Overview. Available at: https://docs.ragas.io/en/stable/concepts/metrics/overview/ [Accessed March 2025].

[AWS 24] Amazon Web Services: What is Retrieval-Augmented Generation (RAG)? Available at: https://aws.amazon.com/what-is/retrieval-augmented-generation/?nc1=h_ls [Accessed March 2025]

[RAG 24] RAGAS Documentation: *Context Precision - Available Metrics*. Available at: https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/context_precision/ [Accessed March 2025].

[RAG 24] RAGAS Documentation: *Context Recall - Available Metrics*. Available at: https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/context_recall/ [Accessed March 2025].

[RAG 24] RAGAS Documentation: Faithfulness - Available Metrics. Available at: https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/faithfulness/ [Accessed March 2025].

[RAG 24] RAGAS Documentation: Answer Relevance - Available Metrics. Available at: https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/answer_relevance/ [Accessed March 2025].