If you find this repository helpful, please consider giving it a star⭐️

# Advanced + Agentic RAG Cookbooks👨🏻‍💻

Welcome to the comprehensive collection of advanced + agentic Retrieval-Augmented Generation (RAG) techniques.

## Introduction🚀

RAG is a popular method that improves accuracy and relevance by finding the right information from reliable sources and transforming it into useful answers. This repository covers the most effective advanced + agentic RAG techniques with clear implementations and explanations.

The main goal of this repository is to provide a helpful resource for researchers and developers looking to use advanced RAG techniques in their projects. Building these techniques from scratch takes time, and finding proper evaluation methods can be challenging. This repository simplifies the process by offering ready-to-use implementations and guidance on how to evaluate them.

[!NOTE] This repository starts with naive RAG as a foundation and progresses to advanced and agentic techniques. It also includes research papers/references for each RAG technique, which you can explore for further reading.

## Introduction to RAG💡

Large Language Models are trained on a fixed dataset, which limits their ability to handle private or recent information. They can sometimes "hallucinate", providing incorrect yet believable answers. Fine-tuning can help but it is expensive and not ideal for retraining again and again on new data. The Retrieval-Augmented Generation (RAG) framework addresses this issue by using external documents to improve the LLM's responses through in-context learning. RAG ensures that the information provided by the LLM is not only contextually relevant but also accurate and up-to-date.

There are four main components in RAG:

**Indexing:** First, documents (in any format) are split into chunks, and embeddings for these chunks are created. These embeddings are then added to a vector store.

**Retriever:** Then, the retriever finds the most relevant documents based on the user's query, using techniques like vector similarity from the vector store.

**Augment:** After that, the Augment part combines the user's query with the retrieved context into a prompt, ensuring the LLM has the information needed to generate accurate responses.

**Generate:** Finally, the combined query and prompt are passed to the model, which then generates the final response to the user's query.

These components of RAG allow the model to access up-to-date, accurate information and generate responses based on external knowledge. However, to ensure RAG systems are functioning effectively, it’s essential to evaluate their performance.

## RAG Evaluation📊

Evaluating RAG applications is important for understanding how well these systems work. We can see how effectively they combine information retrieval with generative models by checking their accuracy and relevance. This evaluation helps improve RAG applications in tasks like text summarization, chatbots, and question-answering. It also identifies areas for improvement, ensuring that these systems provide trustworthy responses as information changes. Overall, effective evaluation helps optimize performance and builds confidence in RAG applications for real-world use. These notebooks contain an end-to-end RAG implementation + RAG evaluation part in Athina AI.

## Advanced RAG Techniques⚙️

Here are the details of all the Advanced RAG techniques covered in this repository.

|  |  |  |  |
| --- | --- | --- | --- |
| Technique | Tools | Description | Notebooks |
| Naive RAG | LangChain, Pinecone, Athina AI | Combines retrieved data with LLMs for simple and effective responses. |  |
| Hybrid RAG | LangChain, Chromadb, Athina AI | Combines vector search and traditional methods like BM25 for better information retrieval. |  |
| Hyde RAG | LangChain, Weaviate, Athina AI | Creates hypothetical document embeddings to find relevant information for a query. |  |
| Parent Document Retriever | LangChain, Chromadb, Athina AI | Breaks large documents into small parts and retrieves the full document if a part matches the query. |  |
| RAG fusion | LangChain, LangSmith, Qdrant, Athina AI | Generates sub-queries, ranks documents with Reciprocal Rank Fusion, and uses top results for accurate responses. |  |
| Contextual RAG | LangChain, Chromadb, Athina AI | Compresses retrieved documents to keep only relevant details for concise and accurate responses. |  |
| Rewrite Retrieve Read | LangChain, Chromadb, Athina AI | Improves query, retrieves better data, and generates accurate answers. |  |
| Unstructured RAG | LangChain, LangGraph, FAISS, Athina AI, Unstructured | This method designed to handle documents that combine text, tables, and images. |  |

## Agentic RAG Techniques⚙️

Here are the details of all the Agentic RAG techniques covered in this repository.

|  |  |  |  |
| --- | --- | --- | --- |
| Technique | Tools | Description | Notebooks |
| Basic Agentic RAG | LangChain, FAISS, Athina AI | Agentic RAG uses AI agents to find and generate answers using tools like vectordb and web searches. |  |
| Corrective RAG | LangChain, LangGraph, Chromadb, Athina AI | Refines relevant documents, removes irrelevant ones or does the web search. |  |
| Self RAG | LangChain, LangGraph, FAISS, Athina AI | Reflects on retrieved data to ensure accurate and complete responses. |  |
| Adaptive RAG | LangChain, LangGraph, FAISS, Athina AI | Adjusts retrieval methods based on query type, using indexed data or web search. |  |
| ReAct RAG | LangChain, LangGraph, FAISS, Athina AI | System combining reasoning and retrieval for context-aware responses |  |

## Demo🎬

A quick demo of how each notebook works:

https://github.com/user-attachments/assets/c6f17961-40a1-4cca-ab1f-2c8fa3d71a7a

## Getting Started🛠️

First, clone this repository by using the following command:

git clone https://github.com/athina-ai/rag-cookbooks.git

Next, navigate to the project directory:

cd rag-cookbooks

Once you are in the 'rag-cookbooks' directory, follow the detailed implementation for each technique.

## Creators + Contributors👨🏻‍💻

## Contributing🤝

If you have a new technique or improvement to suggest, we welcome contributions from the community!

## License📝

This project is licensed under [MIT License](LICENSE)