Artificial Intelligence (Machine Learning & Deep Learning) [Course]

Week 10 – LangChain - Retrieval Augmented Generation (RAG)

[See examples / code in GitHub code repository]

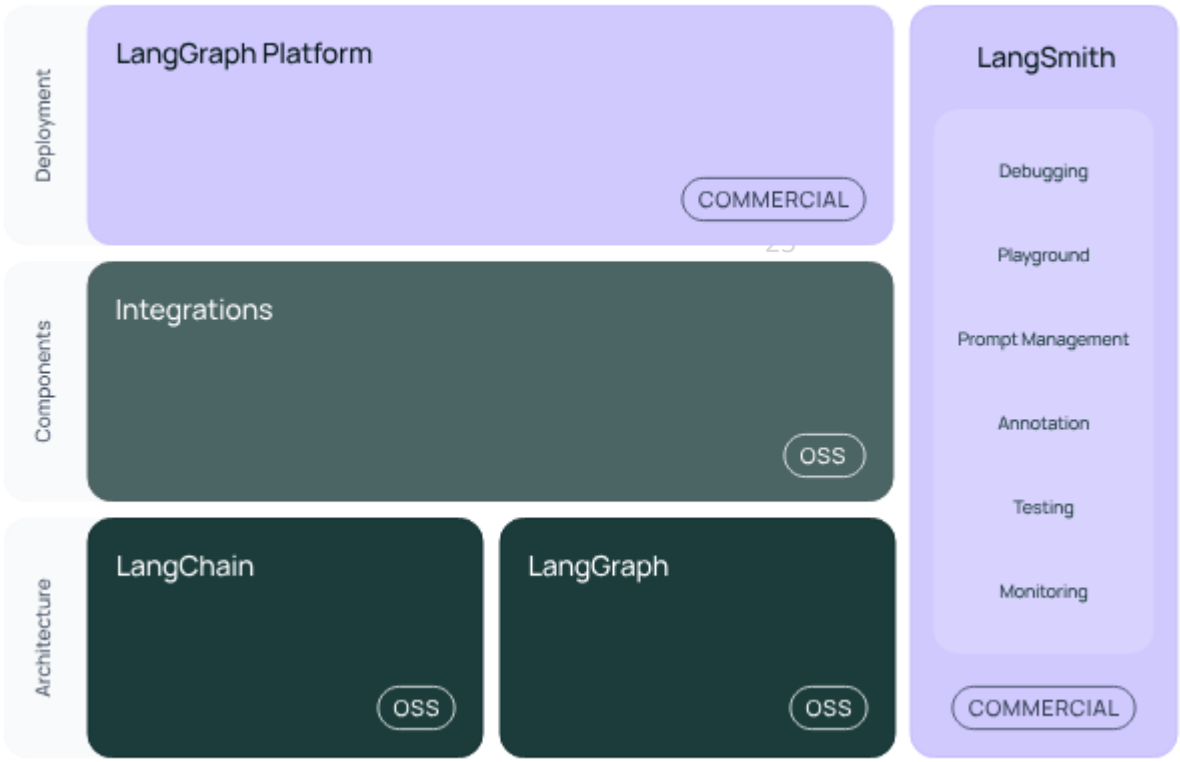
It is not about Theory, it is 20% Theory and 80% Practical – Technical/Development/Programming [Mostly Python based]

LangChain - Foundation

LangChain is a framework for developing applications powered by large language models (LLMs).

LangChain simplifies every stage of the LLM application lifecycle:

- **Development**: Build your applications using LangChain's open-source components and third-party integrations. Use LangGraph to build stateful agents with first-class streaming and human-in-the-loop support.
- **Productionization**: Use LangSmith to inspect, monitor and evaluate your applications, so that you can continuously optimize and deploy with confidence.
- **Deployment**: Turn your LangGraph applications into production-ready APIs and Assistants with LangGraph Platform.

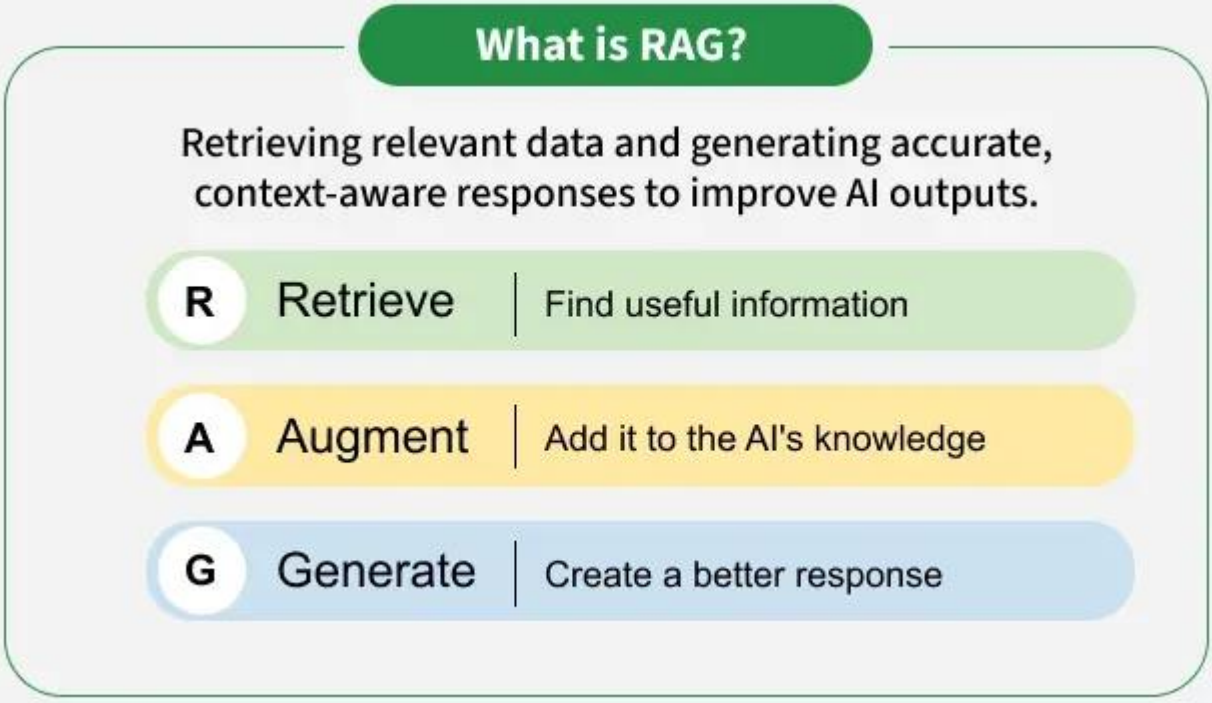


Reference:

https://python.langchain.com/docs/introduction/
https://www.langchain.com/
https://aws.amazon.com/what-is/langchain/

Retrieval-Augmented Generation (RAG) - Foundation

Retrieval-augmented generation (RAG) is an innovative approach in the field of natural language processing (NLP) that combines the **strengths of retrieval-based and generation-based models to enhance the quality of generated text.**



Why is Retrieval-Augmented Generation important?

In traditional LLMs, the model generates responses based solely on the data it was trained on, which may not include the **most current information or specific details required for certain tasks. RAG addresses this limitation by incorporating a retrieval mechanism that allows the model to access external databases or documents in real-time.**

Reference:

https://www.geeksforgeeks.org/nlp/what-is-retrieval-augmented-generation-rag/
https://aws.amazon.com/what-is/retrieval-augmented-generation/
https://cloud.google.com/use-cases/retrieval-augmented-generation?hl=en

LangChain-RAG - Coding – Development Case

Build a Retrieval Augmented Generation (RAG) App

One of the most powerful applications enabled by LLMs is sophisticated question-answering (Q&A) chatbots. These are applications that can answer questions about specific source information. These applications use a technique known as Retrieval Augmented Generation, or RAG.

This tutorial will show how to build a simple Q&A application over a text data source. Along the way we'll go over a typical Q&A architecture and highlight additional resources for more advanced Q&A techniques. We'll also see how LangSmith can help us trace and understand our application. LangSmith will become increasingly helpful as our application grows in complexity.

Practical Development Case Study

Reference:
https://python.langchain.com/docs/tutorials/rag/

Sample Code:

https://colab.research.google.com/github/langchain-ai/langchain/blob/master/docs/docs/tutorials/rag.ipynb
https://github.com/langchain-ai/langchain/blob/master/docs/docs/tutorials/rag.ipynb

Thank you - for listening and participating

- ❑**Questions / Queries**
- ❑**Suggestions/Recommendation**
- ❑**Ideas…..?**

Shahzad Sarwar
Cognitive Convergence
https://cognitiveconvergence.com
shahzad@cognitiveconvergence.com
voice: +1 4242530744 (USA) +92-3004762901 (Pak)