Retrieval-Augmented Generation (RAG)

100% COMPLETE

- RAG basics
- Practice 1: Foundational Text-Based RAG System
- RAG evaluation
- Practice 2: RAG Pipeline Evaluation
- RAG Advanced approaches
- Practice 3: Practice: Hybrid Search + Reranking
- Practice 4: Advanced RAG approaches
- Multimodal RAG
- Practice 5: Multimodal RAG (Phase 5.1 and 5.2)
- Practice 6: Multimodal RAG with ColPali-like approach

Next: Final Assessment

In Progress 50%

Retrieval-Augmented Generation

- Retrieval-Augmented Generation (RAG)
- Final Assessment

Course Tasks 0/1

- Course Evaluation

RAG basics

Grid University

This module aims to provide a comprehensive understanding of Retrieval Augmented Generation (RAG), a technique designed to enhance the capabilities of large language models. The primary objective is for learners to develop a clear understanding of its core architectural principles of RAG and why it is a valuable technique for enhancing large language model systems

Key learning areas:

- **RAG Fundamental Architecture:** explore the roles of retrieval and generative parts, vector embeddings and vector stores. You'll learn how these components operate together to build a cohesive RAG system.
- **Use cases for RAG:** Identify key scenarios and use cases where RAG provides significant advantages. This includes applications requiring factual accuracy, access to proprietary or domain-specific knowledge, and the ability to cite sources.

By the end of this module, you will be able to identify appropriate use cases for applying RAG and design a simple pipeline for your application.

To complete this module, you need to finish the course listed below and review the reading materials.

Fundamental components of a Retrieval Augmented Generation system

This following materials will teach you the fundamental components that form a Retrieval Augmented Generation (RAG) system. We'll explore each key element - from processing user queries and documents into embeddings, storing them in vector stores, fetching relevant information with a retrieval mechanism and understanding how all these parts operate together in a RAG pipeline.

Retrieval-Augmented Generation (RAG)
In this article, we'll explore the limitations of foundation models and how retrieval-augmented generation (RAG) can address these limitations so chat, search, and agentic workflows can all benefit.
READ

Chunking Strategies for LLM Applications
In this post, we'll explore several chunking methods and discuss the tradeoffs needed when choosing a chunking size and method.
READ

Text splitters
Document splitting is often a crucial preprocessing step for many applications. It involves breaking down large texts into smaller, manageable chunks.
READ

What are Vector Embeddings
Vector embeddings are one of the most fascinating and useful concepts in machine learning.
READ

Vector stores
Vector stores are specialized data stores that enable indexing and retrieving information based on vector representations.
READ

How to build RAG pipeline

Examples of how to build RAG pipeline with different frameworks/models:

Build a Retrieval Augmented Generation (RAG) App: Part 1
One of the most powerful applications enabled by LLMs is sophisticated question-answering (Q&A) chatbots.
READ

Basic RAG
Retrieval-augmented generation (RAG) is an AI framework that synergizes the capabilities of LLMs and information retrieval systems.
READ