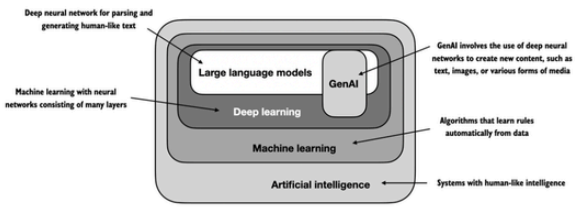
Large Language Model -Tools Set Classifications

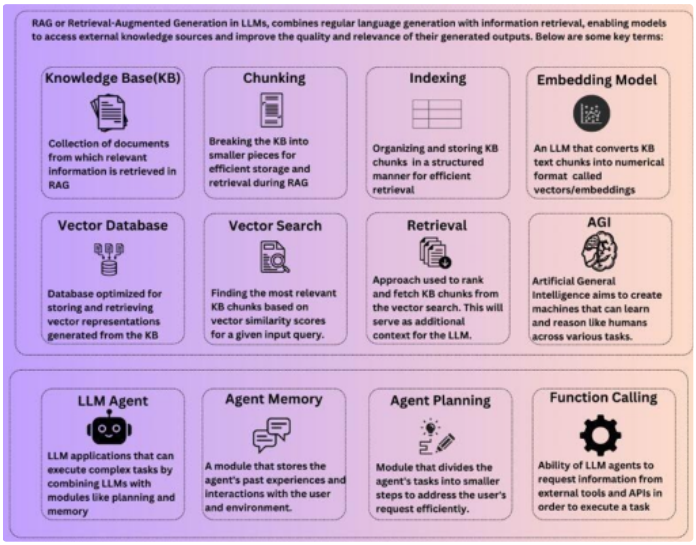
What is an LLM ?

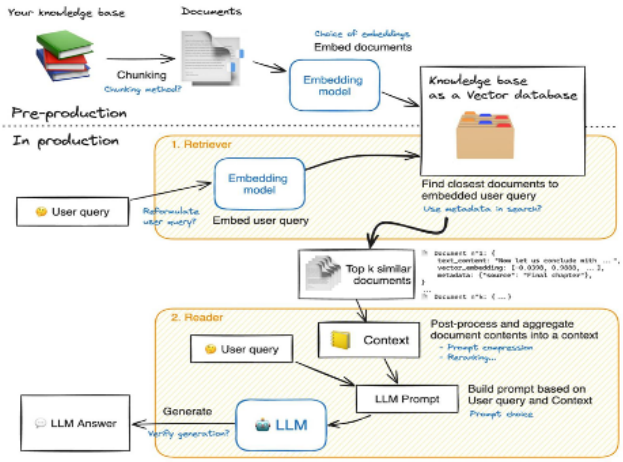
An LLM, a large language model, is a neural network designed to understand, generate, and respond to human-like text. These models are deep neural networks trained on massive amounts of text data, sometimes encompassing large portions of the entire publicly available text on the internet.

LLMs utilize an architecture called the ***transformer*** , which allows them to pay selective attention to different parts of the input when making predictions, making them especially adept at handling the nuances and complexities of human language

The below diagram reads , hierarchical depiction of the relationship between fields suggested, LLM represent of deep Learning (DL) techniques, leveraging their ability to process and generate human-like text.DL is a specialized branch of machine learning (ML) that focus on using multi-layer neural networks.







<Can remove after update>

Knowledge Base(KB)	Chunking	Indexing	Embedding Model

Vector Database	Vector Search	Retrieval	AGI(Artificial General Intelligence)

LLM Agent	Agent Memory	Agent Planning	Function Calling

Custom knowledge base

Custom Knowledge Base: A collection of relevant and up-to-date information that serves as a foundation for RAG. It can be a database, a set of documents, or a combination of both. In this case it's a PDF provided by you that will be used as a source of truth to provide answers to user queries.

- Tools To be used for KB

Chunking

Chunking is the process of breaking down a large input text into smaller pieces. This ensures that the text fits the input size of the embedding model and improves retrieval efficiency.

- Tools To be used for Chunking

Embeddings model

A technique for representing text data as numerical vectors, which can be input into machine learning models. The embedding model is responsible for converting text into these vectors.

We will use Cohere's **embed-english-v3.0** as embedding model.

Indexing

In LlamaIndex terms, an `Index` is a data structure composed of `Document` objects, designed to enable querying by an LLM. Your Index is designed to be complementary to your querying strategy.

LlamaIndex offers several different index types. We'll cover the two most common here.

Vector databases

A collection of pre-computed vector representations of text data for fast retrieval and similarity search, with capabilities like CRUD operations, metadata filtering, and horizontal scaling. By default, LlamaIndex uses a simple in-memory vector store that's great for quick experimentation.

- Tools To be used for Vector Database

User chat interface

A user-friendly interface that allows users to interact with the RAG system, providing input query and receiving output. We have built a streamlit app to do the same.

Query engine

The query engine takes query string to use it to fetch relevant context, reranks the context and then sends them both as a prompt to the LLM to generate a final natural language response. The LLM used here is **Cohere's Command R+** & the reranker used is **Cohere's reranker**! The final response is displayed in the user interface.

Prompt template

A custom prompt template is use to refine the response from LLM & include the context as well:

Conclusion

Basically , we developed a Retrieval Augmented Generation (RAG) application that allows you to "Chat with your documents." and provide tools sets for each phases.