SUPERIOR UNIVERSITY

Submitted By:

Name: Sabiha Liaqat

Roll number   Su92-bsdsm-f23-006

Section: 4A

Task: 11

Subject: pai (lab)

Submitted to: Sir Rasikh

1. LangChain

Definition:
LangChain is a framework designed to simplify the development of applications using large language models (LLMs). It provides tools to integrate LLMs with external data sources, memory, and other components to build powerful AI applications.

Key Features:

- **Chains:** Sequences of calls to LLMs or other utilities.

- **Agents:** Use LLMs to decide actions dynamically.

- **Memory:** Retains context across interactions.

- **Document Loaders & Indexes:** Helps integrate external data.

Use Cases:

- Chatbots with contextual memory.

- Automated content generation pipelines.

- Question-answering systems over custom data. **Difference:**

Unlike standalone LLMs, LangChain orchestrates multiple steps (e.g., fetching data, processing, generating responses) into a cohesive workflow.

2. RAG (Retrieval-Augmented Generation)

Definition:
RAG combines **retrieval** (searching external datasets) with **generation** (LLMs) to produce answers grounded in factual data.

How It Works:

1. Retrieve relevant documents/chunks from a database (e.g., VectorDB).

2. Pass these chunks + the user query to an LLM to generate a context-aware response.

Use Cases:

- Reducing LLM hallucinations by providing evidence.

- Knowledge-intensive tasks (e.g., medical/legal Q&A).

Difference:

RAG enhances LLMs by dynamically fetching external data, while vanilla LLMs rely solely on pre-trained knowledge.

3. LLMs (Large Language Models)

Definition:
AI models trained on vast text data to understand/generate human-like text (e.g., GPT-4, Llama 2).

Key Features:

- **General-purpose:** Can handle diverse tasks (translation, summarization).

- **Zero/Few-shot learning:** Adapt to new tasks with minimal examples.

Limitations:

- Static knowledge (cutoff date).

- Prone to hallucinations. **Difference:**

LLMs are the core engine, while tools like LangChain or RAG build around them to overcome limitations.

4. FAISS (Facebook AI Similarity Search)

Definition:
A library by Meta for **efficient similarity search** in high-dimensional vectors (e.g., embeddings).

Key Features:

- Optimized for fast nearest-neighbor search.

- Supports GPU acceleration.

Use Cases:

- Retrieving similar vectors in RAG pipelines.

- Clustering or recommendation systems. **Difference:**

FAISS is a tool for search, not a database—unlike VectorDBs, which add persistence/management.

5. Vector (Embeddings)

Definition:
Numerical representations of data (text, images) in a high-dimensional space, capturing semantic meaning.

Example:

- "King" – "Man" + "Woman" ≈ "Queen" (in vector math).

Use Cases:

- Search, clustering, input to ML models.

Difference:
Vectors are the data format; FAISS/VectorDBs store/search them.

6. VectorDB (Vector Database)

Definition:
A database optimized to store/query vector embeddings (e.g., Pinecone, Weaviate).

Key Features:

- **Indexing:** Fast similarity search (like FAISS).

- **Persistence:** Unlike FAISS, supports updates/scalability.

- **Metadata storage:** Attach context (e.g., document sources).

Use Cases:

- Long-term memory for AI agents.

- RAG's retrieval step.

Difference:
FAISS is a library; VectorDBs are managed systems with additional features.

7. Generative AI

Definition:
AI that creates new content (text, images, music) rather than just analyzing data.

Examples:

- **Text:** GPT-4.

- **Images:** DALL·E, Stable Diffusion.

- **Multimodal:** Models handling multiple input/output types. **Difference:**

Broad category—LLMs and GANs are subfields under Generative AI.

8. GANs (Generative Adversarial Networks)

Definition:
A neural network architecture where two models compete:

- **Generator:** Creates fake data.

- **Discriminator:** Tries to detect fakes.

Use Cases:

- Photorealistic image generation.

- Data augmentation (synthetic training data). **Difference:**

GANs specialize in unstructured data (images/video), while LLMs focus on text. GANs are a subset of Generative AI.

Summary of Differences

Term	Primary Role	Comparison
LangChain	Framework to chain LLM workflows.	Wraps LLMs + tools (e.g., RAG, VectorDB).
RAG	Combines retrieval + LLM generation.	Enhances LLMs with external data.
LLMs	Base models for text generation.	Core engine for many AI apps.
FAISS	Efficient vector search library.	Faster search vs. brute-force; no persistence.
Vector	Embedding (numeric representation of data).	Fundamental unit for search/generation.
VectorDB	Database for vectors + metadata.	FAISS + storage/management.

Generative AI	Broad field of AI that creates content.	Umbrella term for LLMs, GANs, etc.
GANs	Generates synthetic data via adversarial training.	Specialized for images, unlike LLMs.