

Submitted to: Sir Rasikh Ali

Submitted by: Dua Saeed

Subject: PAI (Lab)

Section: BSAI-4A

Roll number: 042

**. LangChain:**

* **Type**: Framework
* **Purpose**: To build applications that use Large Language Models (LLMs) effectively.
* **Functionality**:
  + Chains together LLMs, APIs, tools, and memory.
  + Supports agents that make decisions using tools (e.g., calculators, retrievers).
  + Integrates with vector databases, prompts, documents, etc.
* **Use Cases**: Chatbots, RAG systems, intelligent agents, custom assistants.

**2. RAG (Retrieval-Augmented Generation):**

* **Type**: AI Technique / Architecture
* **Purpose**: Enhance LLMs by retrieving relevant data from an external source before generating a response.
* **Workflow**:
  + **Retrieve**: Query a knowledge base or document database.
  + **Augment**: Combine retrieved content with user input.
  + **Generate**: Use LLM to generate a response based on this hybrid context.
* **Benefits**:
  + Reduces hallucination by grounding responses in factual content.
  + Allows using **private or real-time data** with LLMs.
* **Use Cases**: Document Q&A, enterprise search assistants, chatbot over PDFs.

**3. LLMs (Large Language Models):**

* **Type**: Deep Learning Models
* **Purpose**: To understand, generate, and process human-like text.
* **Examples**: GPT-4, Claude, LLaMA, Gemini, PaLM.
* **Training Data**: Massive datasets from the internet (books, websites, articles).
* **Capabilities**:
  + Text generation
  + Summarization
  + Translation
  + Code generation
* **Use Cases**: Chatbots, coding assistants, creative writing, virtual tutors.

**4. FAISS (Facebook AI Similarity Search):**

* **Type**: Library/Toolkit
* **Purpose**: Perform fast similarity search over vector embeddings.
* **Key Features**:
  + Handles **billions of vectors** efficiently.
  + Supports indexing methods like IVF, HNSW, Flat, PQ.
* **Use Cases**:
  + Semantic search
  + RAG vector retrieval
  + Recommendation systems

**5. Vector:**

* **Type**: Data Format (Numerical)
* **Definition**: A multi-dimensional numerical representation (embedding) of an object (e.g., text, image).
* **Purpose**:
  + Capture semantic meaning and similarity.
* **Example**:
  + “Cat” and “Dog” might be close in vector space.
* **Use Cases**:
  + Similarity search
  + Clustering
  + Machine learning input

**6. VectorDB (Vector Database):**

* **Type**: Database
* **Purpose**: Store and search **vectors efficiently**.
* **Examples**:
  + Pinecone
  + Weaviate
  + Chroma
  + Milvus
* **Features**:
  + Vector indexing and searching
  + Metadata filtering
  + Integration with RAG pipelines
* **Use Cases**:
  + RAG document search
  + Image similarity search
  + AI assistants with memory

**7. Generative AI:**

* **Type**: Broad AI Category
* **Definition**: AI models that **generate new content** (text, images, audio, code).
* **Models Involved**:
  + LLMs (for text)
  + Diffusion models (for images)
  + GANs (for visual/art)
  + Music generation models
* **Use Cases**:
  + ChatGPT → Text
  + DALL·E → Images
  + GitHub Copilot → Code
  + Sora (OpenAI) → Videos

**8. GANs (Generative Adversarial Networks):**

* **Type**: Neural Network Architecture
* **Definition**: A **two-part model** where a **generator** creates data and a **discriminator** tries to detect fakes.
* **How It Works**:
  + Generator: Tries to fool the discriminator by creating realistic fake data.
  + Discriminator: Tries to detect fake from real data.
* **Common In**: Image synthesis, super-resolution, art generation.
* **Use Cases**:
  + Deepfakes
  + Artistic image creation
  + Image-to-image translation