

1. LangChain

Definition:

LangChain is an **open-source framework** designed to simplify the creation of applications powered by **Large Language Models (LLMs)** like GPT.

Purpose:

It allows developers to combine LLMs with external tools such as:

- Databases (e.g., MySQL, MongoDB)
- APIs (e.g., weather, stock prices)
- File systems (e.g., PDFs, DOCs)
- Agents (for tool use, decision-making)

How It Works:

LangChain provides modules like:

- **Prompt templates** – to structure inputs
- **Memory** – to retain context
- **Chains** – sequences of LLM calls
- **Agents** – for decision-making and tool usage
- **Document loaders + retrievers** – for RAG

Example:

A customer support chatbot that:

1. Understands the user's question.
2. Searches a document database for relevant policy.
3. Uses the LLM to respond in natural language.

2. RAG (Retrieval-Augmented Generation)

Definition:

RAG is a technique that **combines information retrieval** with LLM-based **text generation**.

Purpose:

To allow LLMs to answer questions **based on real-time or external data** rather than just their pre-trained knowledge.

How It Works:

1. A **query** is sent to a retriever (e.g., vector database).
2. Top relevant documents are **retrieved**.
3. Retrieved content is added to the LLM's input.
4. The LLM **generates** a response using both the query and retrieved data.

Example:

Asking “What are OpenAI's 2024 research papers?” → RAG searches a research database and feeds the result to the LLM to generate an answer.

3. LLMs (Large Language Models)

Definition:

LLMs are **deep learning models** trained on massive amounts of text to **understand, generate, translate, and reason in natural language**.

Purpose:

To perform a wide variety of **natural language processing (NLP)** tasks.

Features:

- Pre-trained on large corpora (internet, books, Wikipedia).
- Can perform zero-shot, few-shot, or fine-tuned tasks.
- Popular LLMs: **GPT (OpenAI)**, **LLaMA (Meta)**, **Claude (Anthropic)**, **Gemini (Google)**.

Example:

- Input: “Summarize this paragraph.”
 - LLM generates a coherent summary using learned language patterns.
-

4. FIASS (Few-shot In-context Auto Suggestion System)

Definition:

FIASS is likely a **custom or niche term**, possibly used in research or an organization. Based on its name:


- **Few-shot:** The system learns from a **few examples** provided at runtime.
- **In-context:** Examples are given as part of the **prompt**, not fine-tuned into the model.
- **Auto Suggestion:** The system suggests content based on context.

Purpose:

To suggest context-aware responses or content based on very limited examples.

Example:

Email writing assistant: You type "Dear Dr.," and it suggests a full formal message, learned from just a few prompts.

 *Note: Provide the paper or source if this is a defined system in your curriculum.*

5. Vector**Definition:**

A **vector** is a numerical representation (usually a list of floating-point numbers) of data such as words, images, or sentences.

Purpose:

To enable **mathematical operations** like similarity, distance, clustering, etc.

In AI/NLP:

Vectors capture **semantic meaning** of text. Words with similar meaning have similar vectors.

Example:

- "Paris" – [0.25, -0.10, 0.8, ...]
 - "London" – [0.22, -0.12, 0.79, ...]
 - Cosine similarity between them is high.
-

6. VectorDB (Vector Database)

Definition:

A **vector database** stores vectors and allows fast, approximate **similarity search** among them.

Purpose:

Used in applications like **semantic search, recommendation systems, RAG**, etc.

How It Works:

1. Text is converted to a vector using an embedding model.
2. The vector is stored in the DB.
3. At query time, a similar vector is retrieved based on distance (cosine, Euclidean, etc.).

Examples:

- **FAISS** (Facebook)
- **Pinecone**
- **Weaviate**
- **Chroma**

Use Case:

In RAG, a question is turned into a vector → matched with document vectors → retrieved → used for generation.

7. Generative AI

Definition:

A branch of AI that focuses on **generating new data** (text, images, audio, code) instead of just analyzing or classifying it.

Purpose:

To **create content** that resembles human-made data.

Models:

- **Text:** GPT-4, Claude
- **Images:** DALL·E, Midjourney, Stable Diffusion
- **Audio:** Jukebox, AudioLM

- **Video:** RunwayML, Synthesia

Techniques:

- Transformer models (e.g., GPT)
- GANs
- Diffusion models

Use Case:

Writing essays, generating images from text prompts, composing music, making game assets.

8. GANs (Generative Adversarial Networks)

Definition:

A **generative model** framework where two neural networks (Generator and Discriminator) compete in a game-like setting.

Purpose:

To generate **realistic fake data**, such as images, video, or music.

How It Works:

1. **Generator** creates fake data.
2. **Discriminator** evaluates real vs. fake.
3. Both improve over time—generator gets better at fooling the discriminator.

Example:

- Creating fake faces (e.g., thispersondoesnotexist.com)
 - Generating art, anime characters
 - Deepfake videos
-

Final Summary Table:

Term	Category	Description	Example Use Case
LangChain	Framework	Build LLM-based applications with tools, data, and memory	Chatbots, document QA
RAG	Technique	Retrieve external data + feed to LLM for improved answers	Search-based AI assistant
LLM	Model	Language models that understand and generate text	GPT-4, LLaMA, Claude
FIASS	Likely System/Method	Few-shot auto-suggestion using context	Smart email or code suggestion system
Vector	Data Representation	Numeric form of text/image for machine understanding	Word embeddings
VectorDB	Database	Stores vectors for fast similarity retrieval	FAISS, Pinecone
Generative AI	AI Paradigm	AI that creates new text, images, sound, etc.	ChatGPT, Midjourney, DALL·E
GANs	Model Architecture	Competing neural nets for generating realistic outputs	Deepfakes, image generation, AI art