

Generative AI:

Generative AI is artificial intelligence that can create original content, such as text, images, music, and video. These systems are trained on large datasets of human-generated content and can generate new pieces of original and impressive range.

LangChain:

LangChain is a framework for developing applications powered by language models. There are several main modules that LangChain provides support for. These modules are **Models, Prompts, Memory, Indexes, Chains, and Agents**.

The above modules can be used in a variety of ways. LangChain also provides guidance and assistance in this. Below are some of the common use cases LangChain supports.

Autonomous Agents, Agent Simulations, Personal Assistants, Question Answering, Chatbots, Querying Tabular Data, Code Understanding, Interacting with APIs, Extraction, Summarization, and Evaluation.

Why do we need LangChain?

Imagine you want to use an LLM to answer questions about a specific field, like medicine or law. While the LLM may be able to answer general questions about the field, it may not be able to provide more detailed or nuanced answers that require specialized knowledge or expertise. To work around this limitation, LangChain offers a useful approach where the corpus of text is preprocessed by breaking it down into chunks or summaries, embedding them in a vector space, and searching for similar chunks when a question is asked.

Use Cases of LangChain -

1. **Chatbots:** LangChain Can be used to create chatbots that use LLMs to interact with other tools and do more grounded question-answering or take actions.
2. **Data Augmented Generation:** LangChain allows developers to generate similar examples to a given input, experiment with different prompts, models, and chains.
3. **Tabular Data Querying:** LangChain can be used to query data that is stored in a tabular format.
4. **Summarization:** Developers can use LangChain to summarize longer documents into shorter, more condensed chunks of information.

5. **Interacting with APIs:** Enabling LLMs to interact with APIs is extremely powerful in order to give them more up-to-date information and allow them to take actions.
6. **Personal Assistants:** The main LangChain use case. Personal assistants need to take actions, remember interactions, and have knowledge about your data.
7. **Question Answering:** Developers can answer questions over specific documents, only utilizing the information in those documents to construct an answer.
8. **Evaluation:** LangChain provides prompts/chains for assisting in evaluating generative models.

We will explore more about Interacting with APIs:

There are two methods to interact with APIs **Chains** and **Agents**.

1. **Chains:**

Chains are for relatively simple APIs. Chains are a sequence of predetermined steps. It gives you more control and lets you understand what is happening better.

2. **Agents:**

Agents are more complex and involve multiple queries to the LLM to understand what to do. The downside of agents is that you have less control. The upside is that they are more powerful, allowing you to use them on larger, more complex schemas.