LEARNING GEN AI

- BY SANJANA MCS

LLM’s Frameworks

LLM frameworks LLM frameworks are **software environments and architectures that help develop, train, and deploy large language models (LLMs).** Right now, a vibrant ecosystem of platforms and tools has sprung up, each geared towards helping you interact with your LLM and unlock its full capacity .

Let’s explore some of the top contenders.

- Llama Index
- Langchain
- Autogpt
- Langdock
- Hugging Face agents
- Microsoft Guidance
- Deep speed- MII
- Nvidia Nemo

In this we are going to mainly focus on LLama Index and Langchain

Coming to the frame works to really understand the frameworks we have to understand abstraction.

No matter which language you’re using, coding complex processes require **a lot** of code. LLM frameworks utilize abstractions to replace complex code with simplified code and syntax. Consider it a shortcut that makes your life — and the work of your LLM — much more simple.

Example,

When you make a call, send a message, or take a photo, you don’t need to know how the cellular network works, how the touchscreen detects your touch, or how the camera processes images. These complex operations are hidden from you (abstracted) and are made accessible through simple interfaces like buttons, menus, or apps.

The idea of abstraction is that the processes — no matter how complicated they are — are contained by a component. And these components can then be linked together to create an app.

Frameworks are a library of abstractions. They represent the steps and concepts you need to work with LLMs and make your code more flexible, readable, maintainable, and scalable.

Langchain

Langchain is an open-source framework for developing applications that use large language models. Its tools and API'S make it easier to set up some of the most novel uses of NLP and LLMs. We're talking about chatbots, content summarization, question-answering, and intelligent search — an AI technology that gives you more accurate and relevant online search results.

LangChain supplies your LLM with various data sources and gives it the ability to make decisions on the best way to generate output. Its standardized interface lets you work with any LLM or multiple LLMs at the same time, as well as multiple data sources.

The main value proportion of Langchain can be dived into 3

Components:

- LLM wrappers (allows us to connect to llms)
- Prompt Templates
- Indexes that allows us to extract relevant data for llms

Chains:

- Chains allows us to combine multiple components together
- to solve a specific task and build applications

Agents:

- There are agents that allow llm to interact with external API'S

Core Components of LangChain

Architecture

LangChain's architecture is modular and divided into specific packages, each serving a unique purpose:

- **langchain-core**: Provides core interfaces and components for building applications.
- **langchain**: Includes prebuilt chains and tools for various use cases.
- **langgraph**: Facilitates the orchestration of complex workflows involving AI.
- **langserve**: Handles the deployment of LangChain-based applications seamlessly.
- **LangSmith**: A specialized tool for debugging, monitoring, and fine-tuning LangChain workflows, enabling efficient evaluation and tracing of LLM interactions.

Chat Models and Communication

LangChain supports **chat-based LLMs** that allow structured interactions with users. Key elements include:

- **Messages**: Represent user queries and model-generated responses.
- **Chat History**: Maintains context across multiple user interactions, enhancing continuity.
- **Streaming**: Enables real-time response generation for better user experience.

Tool Integration and Function Calling

LangChain enables LLMs to interact with various external tools for dynamic and enhanced functionality, including:

- **APIs**: To fetch real-time data and perform external operations.
- **Databases**: For executing structured queries and accessing stored information.
- **Search Engines**: For document retrieval and knowledge enrichment.
- **Custom Functions**: Developers can define unique functions tailored to specific requirements.

Retrieval Augmented Generation (RAG)

LangChain incorporates **retrieval techniques** to improve model responses by accessing relevant external information. This involves:

- **Embedding Models**: Converting textual data into vector representations.
- **Vector Stores**: Efficiently storing and retrieving embeddings for large-scale operations.
- **Retrievers**: Identifying relevant documents or data points based on user queries.

AI Agents and Workflow Automation

LangChain enables the development of **autonomous AI agents** capable of performing complex tasks. These agents can:

- Execute **multi-step reasoning** to solve intricate problems.
- Dynamically **call external tools** based on requirements.
- Leverage memory and context to deliver **improved and personalized responses**.

Prompt Engineering and Output Formatting

To optimize LLM performance, LangChain provides:

- **Prompt Templates**: Predefined message formats that guide the LLM for consistent output.

- **Output Parsers**: Structures responses into JSON, tables, or other formats for specific applications.
- **Few-Shot Prompting**: Incorporates examples to enhance the quality of responses.

Evaluation and Testing

Ensuring AI reliability is essential for practical applications. LangChain offers:

- **Callbacks & Tracing**: Tools for tracking and debugging workflows.
- **Testing Frameworks**: Verifying outputs for consistency and accuracy across multiple models.
- **Performance Monitoring**: Evaluating overall efficiency to optimize processes.

Simplified work of LangChain

- **Input**: User provides a query or task (e.g., a question, document, or action request).
- **Data Source Integration**: LangChain connects to data sources (databases, APIs, documents, etc.) for context.
- **Prompt Creation**: Combines user input and retrieved data to generate an effective prompt for the model.
- **Model Interaction**: Sends the prompt to an LLM (e.g., OpenAI, GPT) to generate a response.
- **Processing**: Refines the response using LangChain's tools like chains, agents, and memory.
- **Output**: Delivers the final response or performs the requested task.

Applications of LangChain

LangChain is a versatile framework with applications across various domains:

- **Chatbots & Virtual Assistants**: Used for AI-driven customer support and interactive assistants.
- **Document Search & Summarization**: Powers search engines for retrieving and summarizing information.
- **Code Generation & Debugging**: Helps developers with automated code suggestions and debugging tasks.
- **Workflow Automation**: Automates repetitive and complex tasks with AI-based decision-making.

Challenges and Future Trends

Despite its robust capabilities, LangChain faces several challenges:

- **Scalability Issues**: Managing large-scale applications and datasets can be resource-intensive.
- **Model Hallucinations**: AI models may produce incorrect or misleading information.
- **Ethical Concerns**: Ensuring fairness and mitigating biases remain a priority.

The future of LangChain includes several promising developments:

1. **Better Memory & Personalization**: Enhancing models to deliver more context-aware and tailored responses.
2. **Multi-Modal AI**: Expanding capabilities to include text, images, and video for richer interactions.
3. **Advanced RAG Techniques**: Improving retrieval techniques for faster and more accurate real-time responses.

LLama Index

LlamaIndex (formerly known as GPT Index) is a powerful framework designed to bridge the gap between large language models (LLMs) and external data sources. It simplifies the process of integrating, querying, and managing complex datasets, enabling LLMs to interact with structured and unstructured data effectively.

Workflow of LlamaIndex:

LlamaIndex (formerly known as GPT Index) is a framework designed to connect large language models (LLMs) with external data sources. Its primary purpose is to make structured or unstructured data queryable using LLMs. Below is a simplified workflow:

1. **Data Ingestion**:
    - Input data from various sources, such as CSV files, PDFs, SQL databases, APIs, or raw text.
    - LlamaIndex includes data loaders to facilitate the extraction of information from these formats.
2. **Data Indexing**:
    - The ingested data is organized into **indices** (data structures) that are optimized for retrieval.
    - Popular index types include:
        - **Tree Index**: Organizes data hierarchically for semantic relevance.
        - **List Index**: A simple list of documents.
        - **Keyword Table Index**: Maps keywords to document chunks.

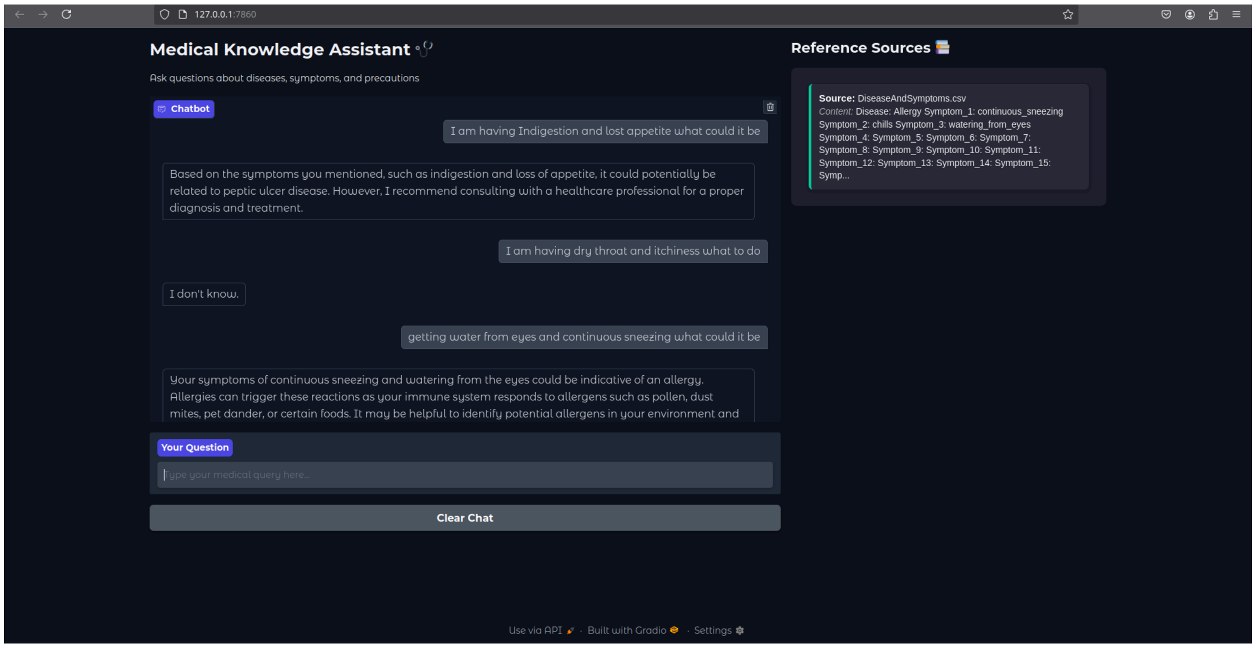
        - **Vector Index**: Uses embeddings to represent data as vectors for similarity-based search.

3. **Embedding Creation**:
    - Data chunks are converted into vector embeddings using a pre-trained embedding model (e.g., OpenAI's embeddings or open-source models like Hugging Face).
    - These embeddings allow the system to compare user queries with indexed data semantically.
4. **Query Execution**:
    - The user submits a query in natural language.
    - LlamaIndex retrieves relevant chunks from the index based on similarity to the query.
5. **LLM Response Generation**:
    - The retrieved chunks are passed to an LLM (like OpenAI's GPT models or local models such as LLaMA or Falcon).
    - The LLM generates a coherent and accurate response based on the retrieved information.
6. **Output**:
    - The final answer is presented to the user, often with references to the source documents.

Differences Between LlamaIndex and Other Frameworks (e.g., LangChain):

In simple terms, LlamaIndex is tailored for creating structured data indices and making them queryable by LLMs, making it ideal for document-heavy tasks like Q&A systems. In contrast, LangChain provides a broader framework for building LLM applications, focusing on dynamic workflows, integration with external tools, and agents. LlamaIndex is efficient for indexing and retrieval, while LangChain offers versatility for various use cases, including dynamic interactions and agent-based tasks.

DEMO MADE:



Medical Knowledge Assistant 🩺

Ask questions about diseases, symptoms, and precautions

I am having Indigestion and lost appetite what could it be

Based on the symptoms you mentioned, such as indigestion and loss of appetite, it could potentially be related to peptic ulcer disease. However, I recommend consulting with a healthcare professional for a proper diagnosis and treatment.

I am having dry throat and itchiness what to do

I don't know.

getting water from eyes and continuous sneezing what could it be

Your symptoms of continuous sneezing and watering from the eyes could be indicative of an allergy. Allergies can trigger these reactions as your immune system responds to allergens such as pollen, dust mites, pet dander, or certain foods. It may be helpful to identify potential allergens in your environment and

Reference Sources 📚

Source: DiseaseAndSymptoms.csv
Content: Disease: Allergy Symptom_1: continuous_sneezing Symptom_2: chills Symptom_3: watering_from_eyes Symptom_4: Symptom_5: Symptom_6: Symptom_7: Symptom_8: Symptom_9: Symptom_10: Symptom_11: Symptom_12: Symptom_13: Symptom_14: Symptom_15: Symp...