Building Apps with RAG and LLM Frameworks

Atul Shukla
atul@kollabia.com

2 Apr, 2024 @ IIITB

Overview

- Limitations of LLM
- How to mitigate LLM limitations
- LLM powered Apps
- LLM Frameworks to build robust LLM Applications
- Demo
    - Building LLM Powered App using RAG and LLM framework

Limitations of LLM

LLMs offer impressive capabilities but suffer from limitations that hinder their effectiveness in certain scenarios. Understanding these limitations is crucial when developing applications.

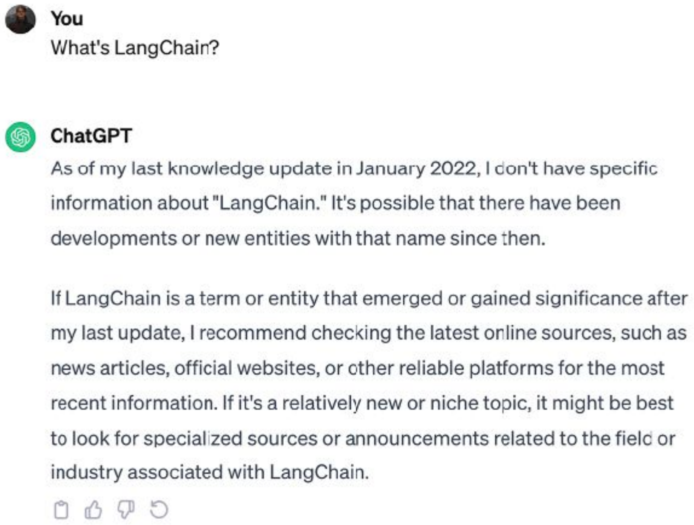
Limitations of LLM

- **Outdated knowledge**: LLMs rely solely on their training data. Without external integration, they cannot provide recent real-world information.
- **Inability to take action**: LLMs cannot perform interactive actions like searches, calculations, or lookups. This severely limits functionality.
- **Hallucination risks**: Insufficient knowledge on certain topics can lead to the generation of incorrect or nonsensical content by LLMs if not properly grounded.

Limitations of LLM

- **Biases and discrimination**: Depending on the data they were trained on, LLMs can exhibit biases that can be religious, ideological, or political in nature.
- **Lack of transparency**: The behavior of large, complex models can be opaque and difficult to interpret, posing challenges to alignment with human values.
- **Lack of context**: LLMs may struggle to understand and incorporate context from previous prompts or conversations. They may not remember previously mentioned details or may fail to provide additional relevant information beyond the given prompt.

Limitations of LLM



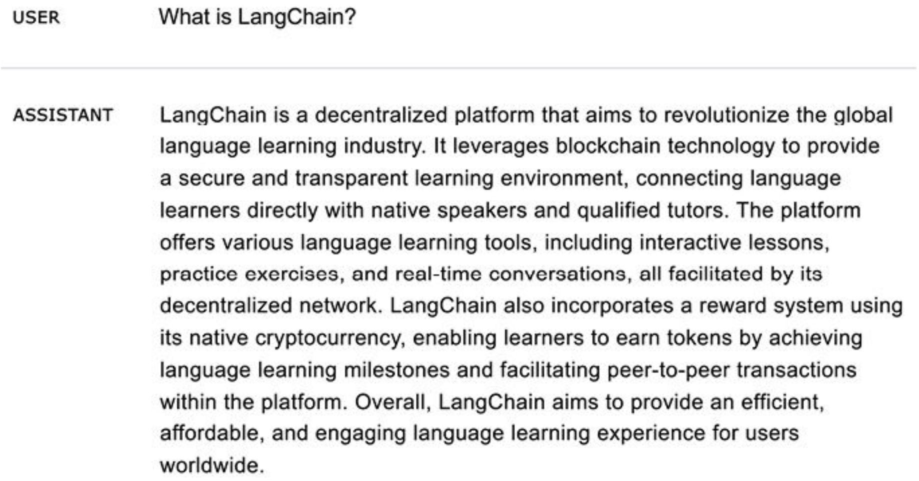
You

What's LangChain?

ChatGPT

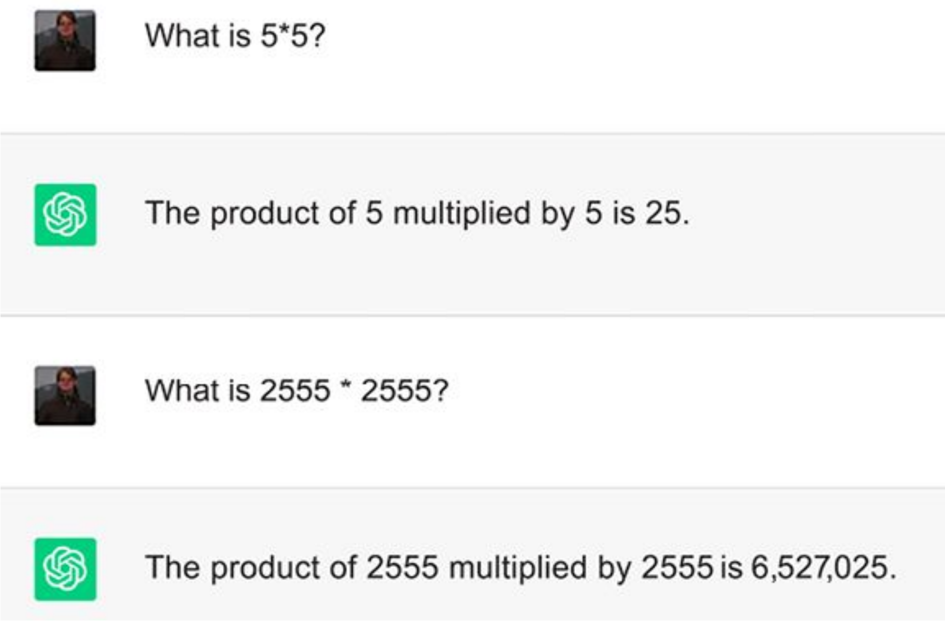
As of my last knowledge update in January 2022, I don't have specific information about "LangChain." It's possible that there have been developments or new entities with that name since then.

If LangChain is a term or entity that emerged or gained significance after my last update, I recommend checking the latest online sources, such as news articles, official websites, or other reliable platforms for the most recent information. If it's a relatively new or niche topic, it might be best to look for specialized sources or announcements related to the field or industry associated with LangChain.



USER	What is LangChain?
ASSISTANT	LangChain is a decentralized platform that aims to revolutionize the global language learning industry. It leverages blockchain technology to provide a secure and transparent learning environment, connecting language learners directly with native speakers and qualified tutors. The platform offers various language learning tools, including interactive lessons, practice exercises, and real-time conversations, all facilitated by its decentralized network. LangChain also incorporates a reward system using its native cryptocurrency, enabling learners to earn tokens by achieving language learning milestones and facilitating peer-to-peer transactions within the platform. Overall, LangChain aims to provide an efficient, affordable, and engaging language learning experience for users worldwide.

Limitations of LLM



What is 5*5?

The product of 5 multiplied by 5 is 25.

What is 2555 * 2555?

The product of 2555 multiplied by 2555 is 6,527,025.

How to mitigate LLM limitations

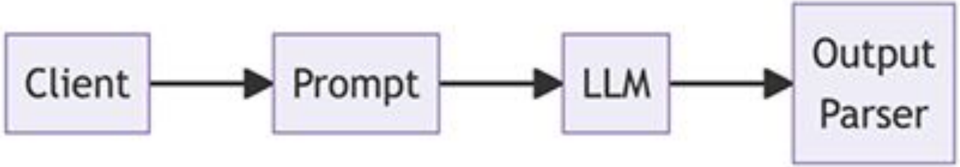
- **Retrieval augmentation**: This technique accesses knowledge bases to supplement an LLM's outdated training data, providing external context and reducing hallucination risk.
- **Prompt engineering**: This involves the careful crafting of prompts by providing critical context that guides appropriate responses.
- **Tools**: Modular interfaces to interact with external systems like databases/APIs.

How to mitigate LLM limitations

- **Memory**: Retains conversation context by persisting conversation data and context across interactions.
- **Fine-tuning**: Training and tuning the LLM on more appropriate data for the application domain and principles. This adapts the model's behavior for its specific purpose.

What is a LLM powered App

- LLM-powered applications combine Language Model (LLM) capabilities with techniques like RAG and other tools to enhance digital experiences.
- Key Components
    - Client Layer, Prompt Engineering Layer, LLM Backend, Output Parsing Layer, Optional Integration with External Services.



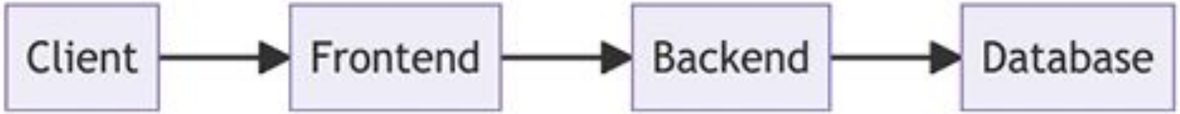
What is a LLM powered App

Components of an LLM App

- **Client Layer:** Collects user input as text queries or decisions.
- **Prompt Engineering Layer:** Constructs prompts that guide the LLM.
- **LLM Backend:** Analyzes prompts and produces relevant text responses.
- **Output Parsing Layer:** Interprets LLM responses for the application interface.
- **Optional Integration with External Services:** Function APIs, Knowledge Bases, Reasoning Algorithms.

Traditional vs. LLM App Architecture

The client layer handles user interaction. The frontend layer handles presentation and business logic. The backend layer processes logic, APIs, computations, etc. Lastly, the database stores and retrieves data.



Example LLM Apps

- **Chatbots and virtual assistants**: These apps use LLMs like ChatGPT to have natural conversations with users and assist with tasks like scheduling, customer service, and information lookup.
- **Intelligent search engines**: LLM apps can parse search queries written in natural language and generate relevant results.
- **Automated content creation**: Apps can leverage LLMs to generate content like articles, emails, code, and more based on a text prompt.
- **Question answering**: Users can ask an LLM app questions in plain language and receive informative answers that are quickly sourced from the model's knowledge.

Example LLM Apps

- **Sentiment analysis**: You can analyze customer feedback, reviews, and social posts using an LLM app to summarize sentiment and extract key themes.
- **Text summarization**: You can automatically generate concise summaries of longer text documents and articles using an LLM backend.
- **Data analysis**: You can use LLMs for automated data analysis and visualization to extract insights.
- **Code generation**: You can set up software pair-programming assistants that can help solve business problems.

LLM Framework

- Facilitates the **development** and **deployment** of robust custom applications powered by Large language models (LLMs), Transformer models, vector search and more.
- Provide a **structured environment** for integrating, managing, and utilizing LLMs to perform various natural language processing tasks efficiently.

LLM Framework

- Supports **features** like prompt engineering capabilities, data processing, retrieval, indexing, querying, and integration with third-party tools and services.
- Enables **orchestration** of processes, data fetching, preprocessing, information retrieval, and **deployment** of LLM-powered applications.
- Include **components** like agents, memory augmentation, and support for GPU acceleration to enhance the functionality and performance of applications utilizing large language models.

Benefits of Using LLM Frameworks

- **Efficiency**
  - Streamline interactions with LLMs
  - Enhance productivity in developing LLM applications
- **Ease of Use**
  - Simplify complex tasks like prompt generation and resource management
- **Performance Monitoring**
  - Monitor and optimize LLM performance effectively

Key components of a LLM Framework

- Models I/O - The interface to the AI Brain
- Data Management - Ingestion/manipulation/indexing/Retrieval
- Pipeline/Workflow/Chain Orchestration
- Memory - Persist information across conversations and workflows.
- Agents - Goal-driven systems that use LLMs to plan actions based on environment observations.
- Tools - Interfaces that agents use to interact with external systems.

Key components of a LLM Framework

These components enable the creation of sophisticated LLM applications that go beyond basic API calls to a single LLM.

Key components of a LLM Framework

- **Models I/O - The interface to the AI Brain**
  - Standard interface for interacting with many different LLMs.
    - Language Model
      - Takes as input a string and returns a string.
    - Chat models
      - A chat model is a language model that uses chat messages as inputs and returns chat messages as outputs (as opposed to using plain text).
      - Chat models with function calling
    - Embedding Models
      - Mainly to be used when comparing text together

Key components of a LLM Framework

- **Models I/O - The interface to the AI Brain**
  - Schemas/Templates (for Dynamic Generation)
    - Text
    - Chat Messages (`SystemMessage, HumanMessage)`
    - Prompts
    - Output Parsers (JSON, XML, CSV etc)
    - Response Synthesizers
  - Communication Modes
    - Sync/Async
    - Batching/Streaming
  - Cache Management
  - Tracking Usage

Key components of a LLM Framework

- **Data Management**
  - Document Loaders
    - Connectors
    - Ingestion
    - Loading
  - Document transformers - Adapt data for model
    - Splitting/Combining
    - Filtering
    - Translation
    - Re-ranking
  - Indexing, Embeddings
    - Index the data into a structure that's easy to retrieve.
    - Usually involves generating vector embeddings

Key components of a LLM Framework

- **Data Management**
  - Storage/Db Connectors
    - Vector databases
      - Stores Embedding
    - Databases
      - Metadata, raw documents etc.
    - Local/Cloud Storage
      - Media Files (audio, video, pdf etc.)
  - Retrievers
    - General interface to return documents based on a query. Can leverage vector stores.
    - Retrieval strategy is key to the relevancy of the data retrieved and the efficiency with which it's done.

Key components of a LLM Framework

- **Memory**
    - Enables the building of stateful, context-aware LLM applications.
    - Persist state/information across conversations, workflows.

Key components of a LLM Framework

- **Pipelines/Chains/Workflows**
  - Compose various components into pipelines/workflows to create powerful and customizable systems.
  - It can include other pipelines as well.
  - Provide runtimes, error handling, failure recovery, retries etc.
- **Agents**
  - Goal-driven systems that use LLMs to plan actions based on environment observations.
  - Agents combine and orchestrate chains. The agent observes the environment, decides which chain to execute based on that observation, takes the chain’s specified action, and repeats.
  - While pipelines compose lower-level modules agents do so by orchestrating chains

Demo

Building LLM Powered App using RAG and LLM framework

Survey of LLM Frameworks and Tools

- Langchain
  - Building applications with LLMs through composability
  - https://github.com/hwchase17/langchain
- Llamaindex
  - Provides a central interface to connect your LLMs with external data.
  - https://github.com/jerryjliu/llama_index
- Haystack
  - Compose applications with LLM Agents, semantic search, question-answering.
  - https://github.com/deepset-ai/haystack
- Dify
  - Open-source framework aims to enable developers (and even non-developers) to quickly build useful applications based on large language models, ensuring they are visual, operable, and improvable.
  - https://github.com/langgenius/dify
- Others
  - LiteLLM - https://github.com/BerriAI/litellm/
    - A simple & light 100 line package to standardize LLM API calls across OpenAI, Azure, Cohere, Anthropic, Replicate API Endpoints
  - GPTCache - https://github.com/zilliztech/GPTCache
    - Creating semantic cache to store responses from LLM queries.

Coming up Next

- Building LLM Apps with Langchain
- Building LLM Apps with LlamaIndex
- How to deploy your Apps
- Production ready LLM Apps
  - Challenges and solutions

Thank You