COE548: LARGE LANGUAGE MODELS

Topic: Agentic Frameworks: LangChain

Outline

- Agentic Frameworks Explained
- Understanding LangChain
- Setting up the Development Environment
- LangChain via Example Codes
  - Defining and Implementing Agents
  - Tool-use
  - Vectorized Datasets
  - Similarity Scores and Vector Search
- Example Tools

Agentic Frameworks

- Agentic frameworks empower applications to act autonomously, making decisions and performing tasks without constant human intervention.
- Relevance in AI:
  - *Enhances the capability of AI systems to perform complex, multi-step tasks.*
  - *Facilitates the integration of various tools and data sources to achieve specific objectives.*
- Real-World Applications:
  - *Virtual assistants, automated research tools, smart home systems, etc.*

Importance of Agentic Frameworks

- Industry relevance: Increasing demand for intelligent, autonomous applications across various sectors.
- Skill development: Equips students with cutting-edge tools and frameworks used in AI and machine learning.
- Innovative potential: Encourages the creation of novel applications that can solve real-world problems efficiently.

What is LangChain?

- A framework designed to simplify the development of applications powered by language models.
  - *Dynamic open-source library that provides developers with a wide range of tools and features to create applications that leverage the capabilities of Large Language Models (LLMs).*
- Facilitates chaining together different components like prompts, tools, and models.
  - *The purpose of LangChain is to establish a connection between these LLMs and external data sources, like personal documents or the internet, allowing developers to access and utilize relevant information.*

What is LangChain?

Flexibility

- As a highly flexible and extensible framework, LangChain allows developers to easily swap components and customize the chain according to their specific requirements. This level of flexibility ensures that developers can tailor their applications for many needs.

Speed

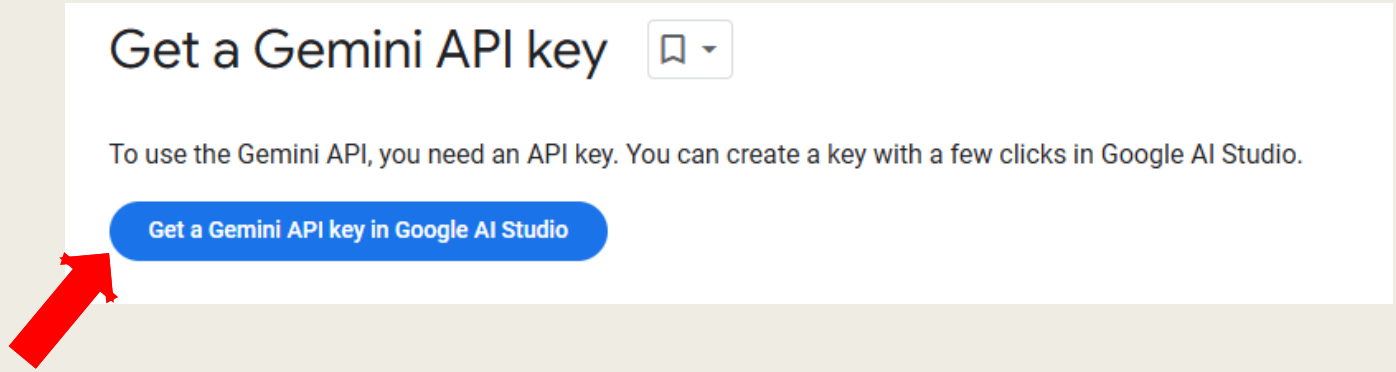
- By incorporating the latest advancements in LLM functionalities, LangChain ensures that users have access to fast and efficient processing capabilities.

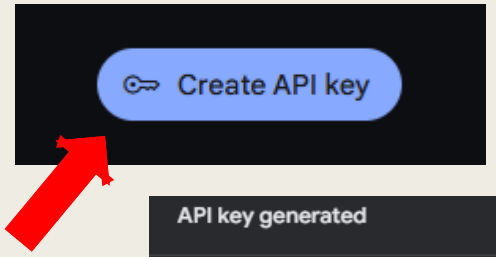
Community

- The LangChain community serves as a valuable resource for developers, offering support and guidance whenever needed. By joining the LangChain community, developers can tap into a network of knowledgeable individuals who can provide assistance and share insights.

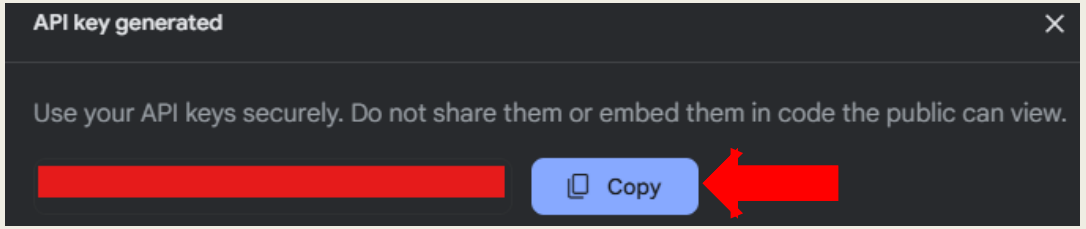
Getting Gemini API Key

- Link: https://ai.google.dev/gemini-api/docs/api-key





Choose the Gemini 1.0 model



Needed Installations for Sample Codes

- `pip install langchain`
- `pip install langchain-google-genai`
- `pip install langchain-core`
- `pip install langchain_chroma`
- `pip install langgraph`
- `pip install chromadb`
- `pip install langchain-community`
- `pip install pypdf`
- `pip install streamlit`

Example Tools
DuckDuckGo

Performs web searches using DuckDuckGo's search engine

- No API key required (free to use)
- Returns search results as text
- Limited to DuckDuckGo's search capabilities
- Example use: search.run("What is the capital of France?")

```
from langchain_community.tools import DuckDuckGoSearchRun

search = DuckDuckGoSearchRun()
```

Example Tools GoogleSearch

Uses Google's Custom Search API

- Requires:
  - *Google API key (paid service)*
  - *Custom Search Engine ID (you create this in Google Cloud Console)*
- More comprehensive search results than DuckDuckGo
- Can be configured to search specific sites
- Example use: google.run("Latest news about AI")

```
from langchain_community.tools import GoogleSearchAPIWrapper

google = GoogleSearchAPIWrapper(
    google_api_key="your_api_key",
    google_cse_id="your_custom_search_engine_id"
)
```

Example Tools
SerpAPI

Provides access to search engine results from Google, Bing, Yahoo, etc.

- Requires SerpAPI key (paid service)
- Returns structured data (including images, news, knowledge graphs)
- More feature-rich than direct Google API
- Example use: serpapi.run("Best restaurants in New York")

```
from langchain_community.utilities import SerpAPIWrapper

serpapi = SerpAPIWrapper(serpapi_api_key="your_api_key")
```

Example Tools
REPL

REPL = Read-Eval-Print Loop

- Executes Python code dynamically
- Useful for: Mathematical calculations; Data processing
- Working with Python libraries
- Example use:

```
python_repl.run("""
import math
radius = 5
area = math.pi * radius ** 2
print(f'Area of circle: {area:.2f}')
""")
```

```
from langchain.tools import PythonREPLTool

python_repl = PythonREPLTool()
```

Example Tools OpenWeatherMap

Provides weather data from OpenWeatherMap

- Requires API key (free tier available)
- Features:
  - *Current weather*
  - *Forecasts*
  - *Temperature, humidity, wind, etc.*
- Example use: weather.run("London,UK")

```
from langchain_community.tools import OpenWeatherMapAPIWrapper

weather = OpenWeatherMapAPIWrapper(openweathermap_api_key="your_api_key")
```

Example Tools NewsAPI

Fetches news articles from NewsAPI

- Requires API key (paid service)
- Features:
  - *Latest news articles*
  - *Search by keyword*
  - *Filter by source, language, date*
- Example use: news.run("artificial intelligence")

```
from langchain_community.tools import NewsAPIWrapper

news = NewsAPIWrapper(newsapi_key="your_api_key")
```