Generative AI

Trainer: Nilesh Ghule

Introduction to LangChain v1 and Agents

- Goal:
  - Raw API scripts to a unified framework for building agents with LangChain.
- Today:
  - What is LangChain v1? ✓
  - How to create agents with cloud and local models? ✓
  - Why the agent-first redesign?
  - How to create minimal agents with cloud and local models?

What is LangChain?

- LangChain is an open-source framework for building LLM-powered applications like chatbots, RAG systems, and agents.
- It provides abstractions for models, prompts, memory, tools, and orchestration, reducing boilerplate and enabling portability across providers.
- Install langchain:
    - cmd> pip install langchain
- Install langchain providers/model integrations:
    - cmd> pip install langchain-groq
    - cmd> pip install langchain-google-genai

Initialize Chat Model

```
from langchain_google_genai import ChatGoogleGenerativeAI

def init_gemini_model():
    api_key = os.getenv("GOOGLE_API_KEY")
    llm = ChatGoogleGenerativeAI(
            model="gemini-2.5-flash",
            api_key=api_key
      )
    return llm
```

```
from langchain_groq import ChatGroq

def init_groq_model():
    api_key = os.getenv("GROQ_API_KEY")
    llm = ChatGroq(
            model="llama-3.3-70b-versatile",
            api_key=api_key
      )
    return llm
```

```
from langchain_openai import ChatOpenAI
def init_local_model():
    api_key = os.getenv("OPENAI_API_KEY")
    llm = ChatOpenAI(
        model="google/gemma-3-4b",
        api_key=api_key
        base_url="http://127.0.0.1:1234/v1"
    ))
    return llm
```

```
from langchain.chat_models import init_chat_model
def init_lanchain_v1_model():
    # use OPENAI_BASE_URL and OPENAI_API_KEY env
    llm = init_chat_model(
        "google/gemma-3-4b",
        model_provider="openai",
        base_url=os.environ.get("OPENAI_BASE_URL"),
        api_key=os.environ.get("OPENAI_API_KEY")
    )
    return llm
```

Thank You!

Nilesh Ghule

<nilesh@sunbeaminfo.com>