



Nome: Fernando Buligon Antunes

Data: 10/05/2025

Introdução ao LangChain

LangChain é um framework open source lançado por Harrison Chase em outubro de 2022, que permite a construção de aplicações que façam o uso de LLMs. Seu uso é de fácil acesso usando Python ou JavaScript. Em sua essência, é uma interface genérica para praticamente qualquer LLM.

Ele permite simplificar a programação de aplicativos envolvendo LLMs através de abstrações, que representam passos comuns e conceitos que são essenciais para trabalhar com LLMs, essas abstrações são “acorrentadas” juntas para criar as aplicações, minimizando o trabalho e tempo gasto fazendo código para executar tarefas complexas de NLP.

A classe padrão de prompts em LongChain formula a composição dos prompts para que não seja necessário colocar o contexto e as buscas manualmente, no próprio template já existem as instruções de como as respostas devem ser.

As “chains” ou, “correntes” em português, são o núcleo do LangChain, elas que combinam LLMs com outros componentes, facilitando a criação de aplicações através da execução sequencial de funções, nesse modelo de execução, a saída de uma função serve como entrada para a próxima.

LongChain possui alguns índices, como o carregador de documentos, que como o próprio nome impõe ele serve para fazer upload de arquivos externos. Outro índice é o vector database, ou em português, banco de dados vetorial, que representam os dados convertendo eles em vetores de embeddings, que são representações numéricas na forma de vetores com um número fixo de dimensões. Também temos os separadores de texto, que são capazes de separar o texto em partes menores, podendo ser configurado da maneira que melhor se enquadrar com o que estiver sendo feito.

Um dos problemas das LLMs é em relação a memória de chats anteriores, ao menos que você passe o histórico no input a LLM teria dificuldade, mas LangChain adiciona memória a aplicação.

E por último, temos os agentes, que são responsáveis por determinar quais ações devem ser tomadas.

Depois são mostrados alguns exemplos de casos de uso do LangChain para aplicações com LLMs, como chatbots, resumos, perguntas e respostas, aumento de dados e agentes virtuais.

Tutorial básico de LangChain

Na parte inicial do vídeo é feita uma introdução sobre o que é LangChain. Depois da introdução, o primeiro passo foi gerar uma chave de API na OpenAI, depois a chave foi guardada dentro de um arquivo .env. Feito isso, foi necessário iniciar um ambiente virtual, nessa parte o método de ativação foi levemente diferente pelo fato de eu estar usando Windows, a primeira linha é necessária por causa das políticas de execução do sistema.

```
PS C:\Users\Fernando Buligon\Documents\fastcamp-lamia\card_09> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
PS C:\Users\Fernando Buligon\Documents\fastcamp-lamia\card_09> .\venv\Scripts\Activate.ps1
(venv) PS C:\Users\Fernando Buligon\Documents\fastcamp-lamia\card_09> 
```



Depois, foi necessário executar três linhas de comando no terminal:

```
pip install langchain pip install openai pip install python-dotenv
```

Depois foi criado um arquivo main.py, nele foi feita uma simples função que passa um prompt e retorna uma resposta, depois foi necessário rodar outra linha de comando:

```
pip install -U langchain-community
```

Ela é necessária pois um dos componentes necessários para importar a OpenAI não vem mais instalado por padrão.

Na época em que o vídeo foi feito, ainda tinha um saldo de teste, em que o usuário tinha acesso a cinco dólares livres para fazer testes com a API, hoje em dia você tem que comprar no mínimo cinco dólares para poder fazer uso. Por isso, optei seguir acompanhando o curso, mas fazendo uso de uma API gratuita da Ollama, que possui funcionamento bem semelhante ao da OpenAI. Foi necessário fazer a instalação do Ollama (pelo próprio site mesmo: [link](#)), depois é preciso abrir um terminal e rodar:

```
ollama run mistral
```

Funcionou certinho, essa foi a saída para o prompt "Adotei um cachorro e estou em dúvida sobre qual nome escolher, consegue me sugerir 5 nomes interessantes por favor?":

```
1. Max - Um nome popular e simpático para cachorros de todos os tamanhos.
2. Bella - É um nome atraente e belo que também funciona bem em cachorros femininos.
3. Cooper - É um nome esportivo e confiável que pode ser usado em cachorros masculinos.
4. Daisy - É um nome feminino adorável para cachorras, é também popular entre as pessoas.
5. Charlie - É um nome tradicional e amigável, que funciona bem tanto para cachorros masculinos quanto para femininos.
```

Memory using LangChain

Foi usado uma API do HuggingFace para criar um chatbot que consegue fazer uso das conversas anteriores para responder a atual, ele acessa o histórico através do buffer, dependendo do modelo tem como configurar o alcance, na imagem abaixo é possível ver o buffer do Conversation Buffer Memory, que funciona bem, mas pode possuir limitações quando a conversa se estende muito, também foi testado o Conversation Buffer Window Memory que é praticamente igual, mas tem um desempenho melhor em conversas grandes, e por último foi mostrado o Conversation Summary Buffer Memory, que tem o diferencial de poder limitar até onde a memória vai, por exemplo, tenho como configurar para que ele só lembre da última conversa.

```
print(memory.buffer) # printa o buffer
```

```
Human: Hello! my name is Fernando
AI: Hello Fernando, my name is Ava. I am a highly advanced artificial intelligence
Human: I study computer science at UTFPR
AI: I am not capable of studying, but I can provide you with knowledge and insights
Human: Whats my name and where do i study?
AI: Your name is Fernando and you study computer science at UTFPR.
```