Nome: Fernando Buligon Antunes **Data:** 13/06/2025

LangChain documentation - Seção chatbot

Foi construído um chatbot com a API do Groq (no total tinham dez opções, optei usar essa pois a maioria das outras são pagas, enquanto essa é de uso livre para fins não comerciais), o modelo usado foi o llama3-8b-8192, e com ele foram feitos diversos testes envolvendo histórico de mensagens, templates de prompt, configuração do histórico da conversa e streaming.

Histórico de mensagens, o modelo não vai ser capaz de te responder com base nas conversas anteriores ao menos que você configure para que ele faça, no chatbot construído esse gerenciamento era feito através do id passado junto com o prompt, ele era responsável por identificar que a sua conversa era contínua, podendo alternar entre sessões livremente.

Templates de prompt, permite dar um contexto maior ao modelo, como por exemplo, no exercício feito, foi passado ao modelo para ele se comportar como um sistema com o objetivo de ser um assistente útil que respondesse todas as perguntas com sua maior capacidade, também foram adicionadas chaves aos prompts, que permitem uma experiência melhor.

Configuração do histórico de mensagens, basicamente seria a personalização de como o modelo deve se comportar em relação às mensagens passadas, por exemplo, qual o range máximo de tokens que ele deve levar em consideração e qual o nível de prioridade deve dar a cada parte.

Streaming, não impacta no funcionamento do modelo, apenas na experiência do usuário, consiste em as palavras serem mostradas para o usuário enquanto são geradas, ver o processo acontecendo fica bem mais confortável para a pessoa atrás da tela do que receber a resposta inteira de uma vez, até porque as vezes pode acontecer da resposta ser meio demorada.

Automating Customer Service using LangChain

No início é comentado sobre como a frase “cliente é que manda” é um mantra passado em qualquer comércio, e que a forma como são tratados é de extrema importância para o funcionamento e prosperidade do serviço. E que para isso, é necessário inovar, pois os métodos tradicionais de atendimento, como FAQs e sistemas fixos, vem se tornando obsoletos.

O artigo propõe mostrar uma abordagem open-source de um chatbot para automatizar o atendimento ao cliente, a metodologia foi separada em quatro partes principais. Coleta de dados, usando técnicas de web scraping com BeautifulSoup para conseguir informações de acesso público. Embeddings, usaram o modelo “hkunlp/instructor-large” da HuggingFace para representar semanticamente os textos. Modelo de linguagem, escolheram o modelo Flan T5 XXL da Google, fizeram a comparação com outros modelos mas esse foi o que mais se destacou em obter conhecimento do espaço vetorial e do histórico do chat, ele se mostrou capaz de compreender o contexto das

mensagens anteriores e usar isso para responder as posteriores, o que é essencial para um chatbot que atende clientes. Integração com plataforma de atendimento a clientes, usaram a API do Gradio para criar uma interface para o usuário que pode ser ativa em qualquer site para interagir com o chatbot.

Depois nos resultados foram montadas três tabelas, cada uma para um dos modelos testados (Flan-T5-XXL, Flan-T5-Base, Flan-T5-Small), na tabela temos as mesmas seis perguntas para os três modelos, qual foi a resposta de cada um e o quão bem o modelo se saiu em cada pergunta. Analisando a performance é possível observar que há uma clara diferença de potencial entre o XXL e os outros dois, o XXL ficou com uma média de 4, Base com 3 e o Small com 1.6.

Conclusão, a automação usando abordagens com LangChain transforma a forma como empresas lidam com o atendimento ao cliente, promovendo maior eficiência, personalização e fortalecimento da imagem da marca.

Construindo um chatbot com LangChain

É feito um resumo sobre o que são LLMs, suas principais capacidades (geração de texto, tradução, resumo, responder perguntas, completar texto e compreensão de linguagem) e também traz exemplos de LLMs populares (GPT-3, Falcon LLM, LLaMA).

Depois introduz o conceito de LangChain, definindo como uma ferramenta poderosa e muito útil que permite desenvolvedores criarem agentes que conseguem analisar problemas e dividir eles em partes menores, tornando capaz superar certas limitações de LLMs. Também é falado sobre Chain, Agentes, memória, ferramentas e um resumo sobre o que foi feito.

Chains, são o núcleo do LangChain, consistem na ligação lógica entre uma ou mais LLM. Agentes, permitem com que o modelo faça uma chamada dinâmica das LLMs com base no prompt, também permitem acesso a fontes externas as LLMs. Memória, permite com que o modelo use as conversas anteriores para responder as atuais. Ferramentas, são funções que podem ser usadas pelos agentes, essas ferramentas podem sem funcionalidades genéricas, outras chains e até mesmo agentes, também foi mostrado como construir uma ferramenta personalizada, no exemplo dado a ferramenta retorna a data atual, depois foi só adicionar ela ao agente.

Foi criado um chatbot que acessa dados do usuário via API, analisa e responde perguntas. Inicialmente usaram um Planner Agent para chamar vários endpoints, mas isso deixava as respostas lentas e pouco naturais, além de não permitir memória nem ferramentas customizadas. A solução foi criar um planner customizado com suporte a memória e ferramentas, integrado a um agente de chat que usa o planner como ferramenta. Assim, o chatbot consegue respostas rápidas, conversa de forma mais natural e usa memória para contexto, melhorando bastante a experiência do usuário.