Faysal Aziz

Advanced Chatbot Development

with Custom and Open-Source LLMsCourse Introduction - 1 day

FA-0028 · 1 day

Agentic & Generative AI

Introduction

As conversational AI becomes increasingly sophisticated, organizations are exploring not only commercial LLM APIs but also open-source models like Llama2, MPT, and NVIDIA's NeMo.

This one-day highly intensive workshop is designed to empower advanced Python developers to harness both proprietary and open-source large language models (LLMs) for building versatile, intelligent chatbots. We will dive deep into OpenAI's API, open-source LLMs, and frameworks like LangChain to design, deploy, and optimize high-performance conversational agents.

With hands-on experience, attendees will learn how to store, fine-tune, and integrate models to deliver robust, context-aware chatbots powered by advanced chaining, role prompting, and retrieval-augmented generation (RAG) techniques.

Learning outcomes

By the end of this course, participants will be able to:

- Use both OpenAI's API and open-source models (Llama2, MPT, NVIDIA NeMo) for chatbot development.
- Store and fine-tune custom LLMs to meet specific conversational needs.
- Implement role prompting, chaining, and retrieval-augmented generation for high accuracy and relevance.
- Utilize LangChain to orchestrate complex conversation flows and manage agent-based interactions.

- Deploy and optimize a fully functional LLM-powered chatbot prototype with a focus on performance and efficiency.

Prerequisites

- Advanced Python proficiency (API handling, asynchronous programming, libraries like NumPy and PyTorch)
- Strong understanding of transformer architecture and artificial neural networks
- Experience with natural language processing (NLP) fundamentals
- Access to high-powered GPUs for model deployment and fine-tuning
- Working knowledge of Docker for containerized model deployment (optional but beneficial)
- Google account (we'll be using Colab Pro)

Training outline

1. Introduction to Modern Chatbot Development with LLMs

1. Overview of conversational AI evolution: From commercial to open-source LLMs
2. Advantages and challenges of OpenAI's API vs. open-source models for production
3. Overview of OpenAI models alongside open-source models (Llama2, MPT, NVIDIA NeMo, etc.)
4. Preparing the lab environment for LLM implementation: setting up GPU configurations

2. Exploring OpenAI's API for Chatbot Development

1. Configuring API keys and setting up requests with Python
2. Understanding OpenAI's model options, endpoints, and rate limits
3. Error handling, token management, and cost considerations
4. Hands-on: Running basic prompts and API calls with OpenAI's models

3. Implementing Open-Source LLMs: Llama2, MPT, and NVIDIA NeMo

1. Introduction to selecting and configuring open-source models
2. Storing models locally and managing GPU resources effectively

   Fine-tuning and customizing open-source models for specific chatbot use cases

   1. Setting up and running a basic fine-tuning pipeline in Python
   2. Comparing Llama2 and MPT model configurations for optimization

4. Hands-on exercise: Deploying a local LLM model for interactive testing

4. Implementing Prompting Techniques to Enhance Chatbot Quality

Role Prompting: Crafting role-specific prompts to maintain consistency and persona in responses

1. Setting roles to refine conversation tone and ensure coherence
2. Practical examples and exercises on role prompting

Chaining Prompts: Creating multi-step prompts to handle complex questions

1. Designing logical flows and maintaining context across multiple queries
2. Hands-on exercise: Developing a conversational chaining example for a realistic interaction scenario

5. Building Agents with LangChain for Interactive and Autonomous Chatbots

1. Introduction to LangChain for managing LLM interactions
2. Setting up LangChain components: Chains, Agents, and Memory
3. Using LangChain Agents to interact with external APIs and dynamic data sources
4. Workshop: Building an agent-based chatbot that pulls in dynamic data (e.g., real-time FAQ or API-driven responses)
5. Integrating multiple agents to handle complex task workflows and enhance chatbot versatility

6. Enhancing Relevance with Retrieval-Augmented Generation (RAG)

1. Overview of RAG and its role in high-context, high-accuracy chatbots

   Setting up document storage and retrieval with vector databases

   1. Using embeddings for document retrieval and ranking
   2. Vector database options and considerations (e.g., Pinecone, Weaviate)
3. Implementing RAG within LangChain: Creating a knowledge-augmented chatbot
4. Hands-on example: Integrating RAG to enable a knowledge-powered conversational agent

7. Building and Integrating Your Custom LLM-Powered Chatbot Prototype

1. Step-by-step integration: OpenAI's API, LangChain, open-source LLMs, RAG, and prompting techniques

2. Using Docker for containerized deployment of custom LLMs
3. Fine-tuning the chatbot for accuracy and performance optimization
4. Setting up logging, monitoring, and debugging tools to track interactions and optimize over time
5. Deploying the chatbot on a server for external or internal user testing

8. Optimization, Troubleshooting, and Performance Enhancement

1. Strategies for managing API costs, reducing latency, and ensuring efficiency
2. Troubleshooting model inconsistencies and common LLM-related issues
3. Model performance tuning: Working with large models on GPU infrastructure
4. Bias management, ethical considerations, and best practices for responsible AI

In this workshop, participants will leave with hands-on experience in developing cutting-edge chatbot solutions, enhanced by both open-source and proprietary LLM technology. With real industry knowledge, this session bridges the gap between research-driven LLM advancements and practical, high-performance AI solutions for real-world conversational needs.

© 2026 Faysal Aziz · Kuala Lumpur