ComponentSoft
open cloud trainings and consultation

AI-435: GenAI Application Development with LLMs (extended version)

Course Length: 40 training hours

Course Description:

The fast pace of development in LLMs and related technologies made it possible to use them even in enterprise grade applications. There are already a few areas where a new generation of LLM-based applications totally redefined applications' capabilities and users' expectations while AI technologies are going to radically change all kinds of other software areas as well.

That's why software developers as well as other IT professionals and technical managers need to understand these technologies and need to have practical skills to use them in their daily work.

Training objectives: At the end of the training participants:

- Recognize the common types of LLM-based applications used today.
- Understand the basic building blocks of modern large language models, how they operate, and the multi-step training process.
- Write simple programs using both open- and closed-source LLMs, either through their own APIs or with popular frameworks like Langchain.
- Understand the main ideas behind prompt engineering, including practical tips and best practices for working effectively with modern LLMs.
- Grasp the fundamental ideas behind RAG (Retrieval-Augmented Generation) systems and apply both basic and more advanced versions in their own LLM applications.
- Understand what LLM-based Agentic Systems are and how to create simple autonomous agents.
- Learn why workflows and multi-agent systems are useful in more complex agentic applications and learn about the most popular open-source agentic frameworks as well as about using the Langgraph framework to build and troubleshoot basic multi-agent systems.
- Recognize the importance of tracking and evaluating LLM applications throughout their lifecycle and get hands-on experience with tools like Langsmith for tracing and assessment.
- Understand when and why fine-tuning open-source LLMs can improve results, along with the technologies involved, such as PEFT (Parameter-Efficient Fine-Tuning) and quantization.

Main topics:

- Introduction to LLM based applications
- Main parts, working and training of LLMs
- Using closed- and open-source LLMs via APIs
- Creating LLM chains with LangChain
- Fast Web Interface Prototyping for LLMs
- Prompt engineering
- Retriever Augmented Generation (RAG)
- LLM-based Agentic Systems
- Multiagent systems and frameworks

- Tracing and Evaluating LLM-based apps
- Fine-tuning Open-source LLM models

Structure: 50% theory, 50% hands on lab exercises

Target audience: Software developers, testers, devops as well as other IT professionals and technical managers with technical backgrounds who want to understand the basic concepts and technologies behind Large Language Models (LLMs) and want to obtain practical skills in LLM application development with the Python APIs of popular closed- and open-source LLMs and open-source frameworks.

Prerequisites: Basic understanding of AI concepts, basic Python programming skills, user experience with ChatGPT or similar chatbots.

This training is part of the AI portfolio of Component Soft which explores essential AI topics:

- AI-110 Intro to Large Language Model (LLMs) and LLM-based apps.
- AI-141: Using Github Copilot as coding assistant
- AI-151: Using Codeium/Windsurf as coding assistant
- AI-161: Using Amazon Q as coding assistant
- AI-434: GenAI Application Development with LLMs
- AI-435: GenAI Application Development with LLMs (extended version)

Detailed Course Outline

PART I. Basic Concepts

Module 1. Introduction to LLM based applications

- Main usage areas of LLM-based applications
- Main types of LLM-based applications
- Building blocks of LLM-based applications

Module 2. LLMs in a nutshell

- Main parts and working of LLMs in a nutshell
- The 3+1 parts of LLM training
- In-context Learning
- Most important base LLM vendors and models
- Lab: Testing text generation of different GPT model generations

PART II. Application Development with LLMs

Module 3. Using closed- and open-source LLMs via APIs

- Using LLMs through APIs
- Typical LLM parameters
- Jupyter Lab basics
- Lab: Doing basic LLM tasks with Jupyter lab notebooks. Using popular closed- and open-source LLMs via the Python APIs

Module 4. Creating LLM chains with LangChain

- What are LLM chains?
- LangChain architecture
- Main Building Blocks: Models, Prompts and Output Parsers
- Building LLM chains from building blocks
- LangChain Memory
- Lab: Using LangChain together with popular closed- and open-source LLMs

Module 5. Fast Web Interface Prototyping for LLMs (Gradio)

- What is Gradio?
- Main features of Gradio
- Building simple GUIs with the ChatInterface class
- Building more complex GUIs with the Block class
- Lab: Building simple and more complex GUIs with Gradio

Module 6. Prompt engineering

- The 4 golden rules of prompt engineering
- 10 Prompting rules of thumb
  - Be concise and give clear instructions
  - Be specific and include relevant details
  - Add positive and negative prompts
  - Define roles for the LLM
  - Define roles for the LLM's audience
  - Provide examples for the solution or response style
  - (one-shot or few-shot prompting)
  - Add relevant context
  - Divide difficult tasks into subtasks (Prompt Chaining)
  - "Let's think step by step" (Chain of Thought)
  - Let the LLM ask questions
- Lab: Prompt engineering tasks

Module 7. Retriever Augmented Generation (RAG)

- What is Retriever Augmented Generation (RAG)
- How do RAG systems basically work?
- Implementation details
- **Lab 1:** RAG Basics
- Advanced RAG techniques
- New directions in RAG
- **Lab 2:** Creating simple and advanced RAG systems

Module 8. LLM-based Agentic Systems

- Motivations for LLM-based Agentic Systems
- Main Features of and Difference between LLM Workflows and Agents
- Main Building Blocks: Functions, Tools, Agents
- The ReAct autonomous agent execution logic
- Implementing Functions, Tools and the ReAct agent execution logic with Langchain and Langgraph
- Lab: Creating and using simple Langgraph autonomous agents

Module 9. Multi-agent systems and Agent Frameworks

- Problems with the ReAct model
- First solution: workflows
- Second solution: multi-agent systems
- Most popular multi-agent frameworks (LangGraph, AutoGen, CrewAI)
- Main building blocks of LangGraph
- Implementing different types of multi-agent systems in LangGraph
- Implementing memory in LangGraph
- Time travel and Human in the loop
- Debugging LangGraph apps with Langgraph Studio
- Lab: Examining and testing typical multi-agent applications in LangGraph

Module 10. Tracing and Evaluating LLM-based apps (Langsmith)

- Why do we need them during development?
- Debugging Langchain-based programs without any monitoring software
- Debugging and evaluation tools for LLM-based apps
- Introducing and Initializing Langsmith
- Langsmith tracing primitives
- Tracing: using Langsmith without and with Langchain
- **Lab:** Langsmith Tracing
- Introduction to Langsmith Evaluation
- Examples of different types of evals
- Langsmith evaluation primitives
- Main steps of Langsmith evaluation
- **Lab:** Langsmith Evaluation

Module 11. Fine-tuning Open-source LLM models

- Typical goals of fine-tuning LLMs
- Difference between prompt engineering and fine tuning
- Which to use?
- Parameter efficient fine-tuning (PEFT) with LoRA and quantized parameters
- The HuggingFace trainer API
- Preparing your dataset
- Creating a fine-tuned model
- Evaluating a fine-tuned model
- Lab: Fine-tuning an open-source LLM