AI-453: Agentic AI Application Development with LLMs (extended version)

Course Length: 24 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, especially agentic AI, and need to have practical skills to use them in their daily work.

Training objectives: At the end of the training participants:
- Understand the basic building blocks of modern large language models, how they operate, and their 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.
- Create simple GUIs for LLM-based applications in Python or JavaScript.
- Understand the main ideas behind prompt engineering, including practical tips and best practices for working effectively with modern LLMs in chatbots and agentic applications.
- Grasp the fundamental ideas behind RAG (Retrieval-Augmented Generation) systems and apply both its basic and more advanced versions in LLM-based agents.
- Understand the motivations for and the two main types of LLM-based agentic systems (workflows and autonomous agents) as well as the key components and the way of working of autonomous agents.
- Learn why workflows, multi-agent systems and deep agents are useful in more complex agentic applications and learn about the most popular agentic frameworks.
- Recognize the importance of observing (tracing) and evaluating LLM-based applications throughout their lifecycle and get hands-on experience with tools like LangSmith for tracing and evaluation.

Main topics:
- 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
- Retrieval Augmented Generation (RAG)
- LLM-based Agentic Systems
- Workflows, Multi-agent Systems and Agentic Frameworks (optional)
- Tracing and Evaluating LLM-based apps

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, such as:

- AI-101: Intro to GenAI with Large Language Model (LLMs) and LLM-based apps.
- AI-242: Using Github Copilot and spec-kit for agentic coding and spec-driven development (SDD)
- AI-262: Using Amazon Kiro for agentic coding and spec-driven development (SDD)
- AI-452: Agentic AI Application Development with LLMs
- AI-453: Agentic AI Application Development with LLMs (extended version)

Detailed Course Outline

PART I. Basic Concepts

Module 1. Main parts, working and training of LLMs
- 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 2. Using closed- and open-source LLMs via APIs

- Using LLMs through APIs
- Typical LLM parameters
- Jupyter Lab basics
- **Lab:** Using popular closed- and open-source LLMs via the Python APIs

Module 3. App. development frameworks, LangChain
- LLM app. development frameworks
- Main features of Langchain
- Langchain components
- LangChain Memory
- **Lab:** Using LangChain together with popular closed- and open-source LLMs

Module 4. Fast Web Interface Prototyping for LLMs in Python and Javascript
- Creating web UI in JavaScript using coding agents
- Python based web GUI frameworks: Gradio and Streamlit
- 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 5. Prompt engineering for chatbots and agents

- 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 LLM ask questions
- **Lab:** Prompt engineering techniques

Module 6. Retrieval Augmented Generation (RAG)

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

Module 7. 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
- **Lab:** Creating and using simple LangChain autonomous agents

Module 8. Workflows, Deep Agents, Multi-agent systems and Agentic Frameworks

- Problems with the ReAct model
- First solution: workflows
- Second solution: multi-agent systems
- Third solution: advanced (deep) agents
- Most popular agentic frameworks
- **Lab**: Examining a deep-research app. using deep agents

Module 9. Observing (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