

AI-434: GenAI Application Development with LLMs

Course Length: 32 training hours

Course Description:

The fast pace of development in LLMs and related technologies made it possible to use them reliably 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.

Consequently, software developers of all tribes 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:

- get to know the typical types of LLM-based applications,
- know the basic building blocks, operation and multi-step training of modern LLMs,
- can write simple programs using closed- and open-source LLMs via their own API or through Langchain, the most popular open-source LLM development framework,
- understand the concepts behind RAG systems and can use their basic and more advanced forms in their LLM-based applications
- understand the concepts of LLM-based Agentic Systems and can write simple autonomous agents
- understand the importance of tracing and evaluation of LLM-based applications throughout their entire lifecycle and can use tracing and evaluation tools such as Langsmith

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
- Workflows, Multi-agent systems and Agentic Frameworks (optional)
- Tracing and Evaluating LLM-based apps

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

- AI-110: Intro to GenAI with Large Language Model (LLMs) and LLM-based apps.
- AI-151: Using Codeium as coding assistant
- AI-434: GenAI Application Development with LLMs

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

Target audience: Software developers and other IT and technical professionals as well as 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 frameworks.

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

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. 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 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 closedand 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 2 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
 - o Define roles for the LLM's audience
 - Provide examples for the solution or response style
 - o (one-shot or few-shot prompting)
 - \circ Add relevant context
 - o Divide difficult tasks into subtasks (Prompt Chaining)
 - o "Let's think step by step" (Chain of Thought)
 - \circ 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
- 6.4. Advanced RAG techniques
- 6.5. 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. Workflows, Multi-agent systems and Agentic Frameworks (optional)

- Problems with the ReAct model
- First solution: workflows
- Second solution: multi-agent systems
- Most popular multi-agent frameworks (LangGraph, AutoGen, CrewAI)
- Main features of 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 evalsLangsmith evaluation primitives
- Main steps of Langsmith evaluation
- Lab: Langsmith Evaluation