A great way to learn about GenAI is to build a project. Check below for a description of different projects and a list of courses that can help you with theory and example code for each project.

Build your own Chatbot

A chatbot project can help solidify your understanding of a few key topics – and is fun to build!

Training: LLMs have two phases of training. Pre-training, where a model is trained to predict the next word, and Fine-Tuning which tunes a model to a particular task. For chat models, that task is to be conversational.

Memory: On its own, an LLM does not have a memory beyond what is presented in a prompt. To remember the turns of a conversation, you can add memory.

Guardrails: If you want to use your chatbot in production, it should be monitored to ensure safe, non-toxic conversations.

Chatbot Courses

ChatGPT Prompt Engineering for Developers: The first lecture of this course will describe the basics of an LLM and the difference between a pre-trained and fine-tuned model. The last lecture, ‘Chatbot’, will show you how to build a chatbot from scratch and even includes a GUI to implement a usable pizza-bot! The lectures in between provide insightful examples of prompting.

Langchain for LLM Application Development: A framework such as Langchain can make a project more robust by providing things like LLM independence, retry logic, tracing, and other features. There is a cost in time to learn about the framework. In this course, you learn to use the Langchain framework and, in lesson 2, you will learn about different types of conversational memory for your chatbot.

Building Systems With ChatGPT: The “Moderation” lecture in this course describes how to use the OpenAI moderation tool to ensure safe conversations. If you take the whole course, the last few lessons will build a question-answering chatbot!

There are many more short courses that will build and use a chatbot interface, but this list should provide a good foundation.

Retrieval Augmented Generation: Question Answering Over Documents

RAG over documents has two phases. In the first phase the documents are processed and installed in a database, often a vector database. The second phase is retrieving the data as shown below:

Let’s go over what these steps are and then below what courses you can take to get up to speed on them. The explanations will be brief. You can take a course to learn more!

Extract: Documents come in all sorts of file formats (.doc, .pdf, etc.) and contain all sorts of data formats (text, tables, images, movies). These must be extracted and put into a format that can be processed by the next stages.

Chunking: Text data is broken into smaller chunks – a process inventively named ‘chunking’.

Embedding: Converting a chunk into a ‘dense vector’ that represents the meaning of the text.

Loading: Adding the embedding and original data to a database.

Database: The database is going to provide storage for the embedding and data. Often these are vector databases due to the embedding, but graph databases and traditional databases are also used.

Query

Embedding: The query is converted to a dense vector using the same embedding model.

Retrieval: The stored and query vectors represent meaning, so retrieval is the process of finding the k entries in the database that are ‘closest’ to the query vector. Lots of details here!

k results are provided to an LLM which uses them to form an ‘augmented’ response.

RAG Courses

The courses you might find useful may depend on what stage you are at. You may want general or specific information.

General Overview

These courses provide an overview of the RAG process. An easy way to get started is to use a framework such as Langchain or LlamaIndex.

LangChain: Chat with Your Data uses the LangChain framework to build a RAG pipeline. It provides a nice overview, discusses all the steps and in the end builds a pipeline and GUI you could use in your applications.

Building and Evaluating Advanced RAG Applications uses the LlamaIndex framework. It also provides a nice overview and builds a RAG application. In addition, it discusses evaluation techniques with the TruEra evaluation tool set.

Building Agentic RAG with LlamaIndex uses the LlamaIndex framework to implement an agentic version of RAG. This is a little more advanced.

Pipeline Specifics

You may want more details on some of the steps above. Many courses have focused on a particular aspect of the pipeline.

Preprocessing Unstructured Data for LLM Applications: This course discusses “Extraction”. Ingesting data from many sources and formats and preparing them for embedding.

Embedding:

Large Language Models with Semantic Search This course covers how to use an embedding model to create dense vectors and vector similarity.

Understanding and Applying Text Embeddings This course, taught with Andrew Ng, discusses what embeddings are, how they are created and how to use them in RAG and other settings.

Vector Databases

Vector Databases: from Embeddings to Applications This course describes how vector databases work, in particular, it describes the Approximate Nearest Neighbor (ANN) algorithm used in many vector databases, Hierarchical Navigable Small World, or HNSW. It demonstrates the Weaviate DB.

Advanced Retrieval for AI with Chroma This course covers some more advanced retrieval techniques and demonstrates the Chroma DB.

Building Applications with Vector Databases This course covers RAG and other applications and demonstrates the Pinecone DB

Mixed Databases

Knowledge Graphs for RAG This course demonstrates RAG applications using the Neo4j graph database which stores objects and their relationships.

Generating Images

You may have used DALL-E from OpenAI or other image generation tools and wondered how these models work. A good way to find out is to build your own!

Image Generation Courses

How Diffusion Models Work: This course will take you through the steps to build and train your own diffusion model.

Fine-tune your LLM

LLMs are trained on publically available data for general use cases. If your application requires unique knowledge, like a profession-specific vocabulary or a distinct personality, you may turn to fine-tuning to accomplish this.

Fine-tuning Courses

Finetuning Large Language Models: This course covers fine-tuning from model selection to training and deployment. This will use the Lamaini LLM-serving service to train and host your model.

Efficiently Serving LLMs: LLM-service services provide the GPUs needed for training and hosting. This course, from Pedibase, describes some techniques used to serve LLMs efficiently. It also gives an overview of LoRa – Low Rank Adaption, a parameter-efficient approach to fine-tuning as well as quantization – a means of reducing the memory used by your model.

Selecting a model to train:

Open Source Models with HuggingFace: The first lesson reviews how to find and select open-source models

Prompt Engineering with Llama-2 and Getting Started with Mistral: These courses describe two popular open-source model families that are frequently used as a base for fine-tuning

Research Agent

Increasingly, LLMs are being used in Agents to perform a sequence of actions using tools. A research agent will develop a research plan, search for sources, and write a report. These courses provide the building blocks needed to build your own agents. It would be a good idea to have previously tried out the chatbot and RAG projects.

A building block of your research agent is tools. Here are two courses on ‘function calling’. This is the LLM capability that enables tool development.

Research Agent Courses

Tools Courses

Functions Tools and Agents with Langchain: The first lesson will teach you to use OpenAI function calling. You then create tools using LangChain. The final lesson, building a conversational agent, is not required. You will learn that in the Agent Courses below.

NexusFlow Function Calling: Coming Soon, This course will teach you to use a small LLM that has been fine-tuned to be an expert in function calling – ideal for building agentic tools.

Having mastered tools you can now build an agent. You can use an agentic framework such as AutoGen or CrewAI that specialize in Agent tasks. Or, you can use more general frameworks such as LangChain, LlamaIndex, or Haystack (coming soon) to build a research agent.

Agent Courses

AI Agentic Design Patterns with Autogen: Autogen is an agentic framework that specializes in agentic tasks. This course provides multiple project examples: Customer Onboarding, Blog-post writing, Tool Use (important for a research agent), Financial Analysis, and Report generation.

Multi AI Agent Systems with crewAI: Crew is a multi-agent framework. This course shows you how to build many projects including a research agent described in lesson 3.

AI Agents in LangGraph: LangGraph allows you to build your own agentic dataflows. It is a bit more effort than a purely agentic framework but allows you fine-grain control of dataflows. This course covers tool use and finishes by building a research agent. It uses Tavily, a search engine built for agents.

Building Agentic RAG with LlamaIndex: This course builds a RAG research agent that can be used to research local documents. LlamaIndex also supports tools like Serper and Tavily to add web search.