PESU I/O Course Plan

Course Name:

RAGs to Rich AIs

with LLM Agents

By:

Samarth P and Vyoman Jain

RAGs to Rich AIs – Course Plan

Instructor Name: Samarth P and Vyoman Jain

SRN: PES2UG22CS495 and PES2UG22CS672

Branch: CSE

Semester: V

Course Duration: 30 Hours (20 hours of in-person mentoring + 10 hours of self learning)

Prerequisites for the course:

1. Basic knowledge of Python

Deliverables from the course:

1. Understand what LLMs are and how they can be used for different real world applications
2. Understand what Retrieval Augmented Generation (RAG) is and why it is used
3. Learn about and implement the different types of RAG - naive, advanced, agentic and Graph using LangChain / LlamaIndex / Crew.ai
4. Have a working idea of what happens in a typical LLM pipeline application enhanced with RAG and what production-level codebases look like
5. Recognize the strengths and weaknesses of different types of RAG and where current research is being concentrated

Final Project:

Students should choose a topic of their choice and select corresponding data sources to build and implement an agentic RAG pipeline.

Video Shoot Content:

Week 1: This video introduces a Generative AI, LLMs and explains how to use them for real-world applications and the need for Retrieval Augmented Generation.
Week 2: Overview of RAG architecture and implementation of naive RAG.
Week 3: Advantages of Agentic RAG and implementation using LlamaIndex / LangChain.
Week 4: Overview of emerging RAG architectures - multi-agent RAG, GraphRAG, HybridRAG

Day 0 – Student Onboarding

1. Introduce ourselves and learn about our students
2. Provide a brief introduction to the world of Generative AI covering what LLMs are, an overview of the Transformer architecture and current applications.
3. Task: Explore different popular LLMs out there and setup either Ollama for running open-source models locally (if system is good enough), else set up a free Groq account for making API calls to hosted models

Day 1 – What is RAG and why is it used?

1. Topics to be taught:

- Understanding why LLMs need RAG or other techniques like Fine-tuning, context-caching
- Introduction to Retrieval Augmented Generation
- Overview of the RAG architecture

2. Tasks to be completed:

- Explore the resources (papers, code, blogs, videos) shared by us.

Day 2 – How RAG Works

1. Topics to be taught:

- Explanation of how RAG works
- Introduction to the different components of RAG architecture
- Types of RAG architecture

2. Tasks to be completed:

- Small tutorial and best practices for Python usage in production

3. Weekly Assignment 1:

- Quiz on topics learnt so far and on resources shared by us

Day 3 – Deep Dive into RAG Architecture Part 1: Preprocessing:

1. Topics to be taught:

- Data collection/loading for RAG by scraping / parsing documents
- Chunking of data for indexing
- Indexing of data and introduction to embedding

2. Tasks to be completed:

- Explore LangChain and LlamaIndex frameworks

Day 4 – Deep Dive into RAG Architecture Part 2: Retrieval and Vector DBs:

1. Topics to be taught:

- Understanding what embedding data actually means and demonstration using Jina.AI
- Introduction to Vector databases and how they work
- Understanding a little bit of the Maths behind how retrieval of context in vector DBs works and demonstration using Pinecone

2. Tasks to be completed:

- Implement semantic search for a dataset shared by us

Day 5 – Deep Dive into RAG Architecture Part 3: Generation

1. Topics to be taught:

- Prompt Engineering for RAG
- Context passing to the LLM
- Implementation of a naive RAG application in class to help with the weekly assignment

2. Weekly Assignment 2:

- Implement any simple application using naive RAG

Day 6 – Intro to Advanced RAG Part 1: Pre-Retrieval Techniques

1. Topics to be taught:

- Query Rewriting / Transformation
- Different chunking strategies - fixed-length, recursive and semantic chunking
- Query Routing to different indices

2. Tasks to be completed:

- Implement some of these techniques in your naive RAG application and observe the difference in output quality

Day 7 – Intro to Advanced RAG Part 2: Post-Retrieval Techniques

1. Topics to be taught:

- Reranking of retrieved chunks for optimized context passing to the LLM
- Comparing different reranking models and how to choose one of them
- Context compression and filtering for improved efficiency and performance

2. Tasks to be completed:

- Implement some of these techniques in your naive RAG application and observe the difference in output quality

Day 8 – Intro to Agentic RAG applications:

1. Topics to be taught:

- Limitations of naive RAG and how Agentic applications can benefit us
- Introduction to Agentic RAG architectures
- Demonstration of an agentic application

2. Tasks to be completed:

- **Final Project Discussion** - Topic selection and Team Formation

Day 9 – Hands On: Using LlamaIndex / LangChain to build an agentic application with tool calling

1. Topics to be taught:

- How to use frameworks like LlamaIndex and LangChain

2. Tasks to be completed:

- Using LlamaIndex or LangChain to build an agentic RAG application

3. Weekly Assignment 3:

- Build your own web-scraping agents for up-to-date Question Answering

Day 10 – Hands On: Using Crew.ai to build multi-agent systems + doubt session

1. Topics to be taught:

- Introduction to multi-agent frameworks like Crew.ai and Autogen
- Tutorial on how to build a multi-agent system using Crew.ai

2. Tasks to be completed:

- Implement a Research assistant which helps people with literature reviews using Crew.ai

Day 11- Graph RAG & other emerging architectures

1. Topics to be taught:

- Introduction to GraphRAG - an new and open-source framework from Microsoft
- Brief overview of other emerging architectures and techniques such as HybridRAG, RAFT, LangGraph

2. Tasks to be completed:

- Work on the final project and ask any doubts in class

Day 12 – Project Presentation:

Tasks to be completed:

- Students present their final projects
- All projects are evaluated by us