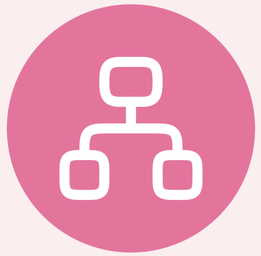
Machine Learning

Engineer in the Generative AI Era

Course Structure

10 WEEKS
2-HOUR
WEEKLY LECTURES



WEEKLY PROJECTS:
MEDIUM & CHALLENGING
LEVELS

FOCUSED ON DATA
ENGINEERING FOR
LLMS

EVERYTHING BUILDS
TOWARD THE FINAL
RESEARCH AGENT

Course Schedule

Weeks 1–3: Data Engineering

Week	Topic	Lecture Themes	Project
1	Intro to LLMs & Prompt Engineering	Generative AI & agents, LLM capabilities, prompting techniques (CO-STAR, JSON/XML output)	Prompt design for research agent, using CO-STAR and structured formats
2	LLM Architecture & Training Lifecycle	Transformers, hallucination, SFT/DPO/PPO, test-time scaling, pretraining data requirements	Run inference with local LLMs (e.g., LLaMA 3/4), evaluate with designed prompts
3	Pretraining Data Collection & Extraction	Web scraping, OCR (Tesseract/Surya), ASR (Whisper), data cleaning/filtering (PII removal, deduplication)	Scrape arXiv, OCR PDFs, filter & clean data for pretraining

Course Schedule

Weeks 4–7: Introduction to AI & Model Training

Week	Topic	Lecture Themes	Project
4	Retrieval-Augmented Generation (RAG)	Embeddings, chunking, vector DBs, LangChain, RAG workflows	Build a RAG pipeline to augment an LLM with external knowledge
Project insight I		Review for the project idea	
5	Supervised Fine-Tuning (SFT) I	Full vs. LoRA fine-tuning, ChatML format, TRL/Deepspeed	Apply LoRA and full fine-tuning using public datasets, explore overfitting
6	Supervised Fine-Tuning (SFT) II	Synthetic data, quality checks, LLM-as-judge, data diversity ablation	Generate synthetic SFT data, tune with mixed datasets, perform ablation studies
7	Model Alignment	RLHF, DPO/PPO, reward modeling, data labeling platforms	Build a labeling tool in Gradio, label preference data, run a DPO alignment experiment
Project insight II		Decide what project you will work on	
8	Hallucination, Jailbreak, and Ethics	Safety alignment, jailbreak cases, hallucination prevention	Try jailbreaking models, simulate hallucination, explore safety datasets

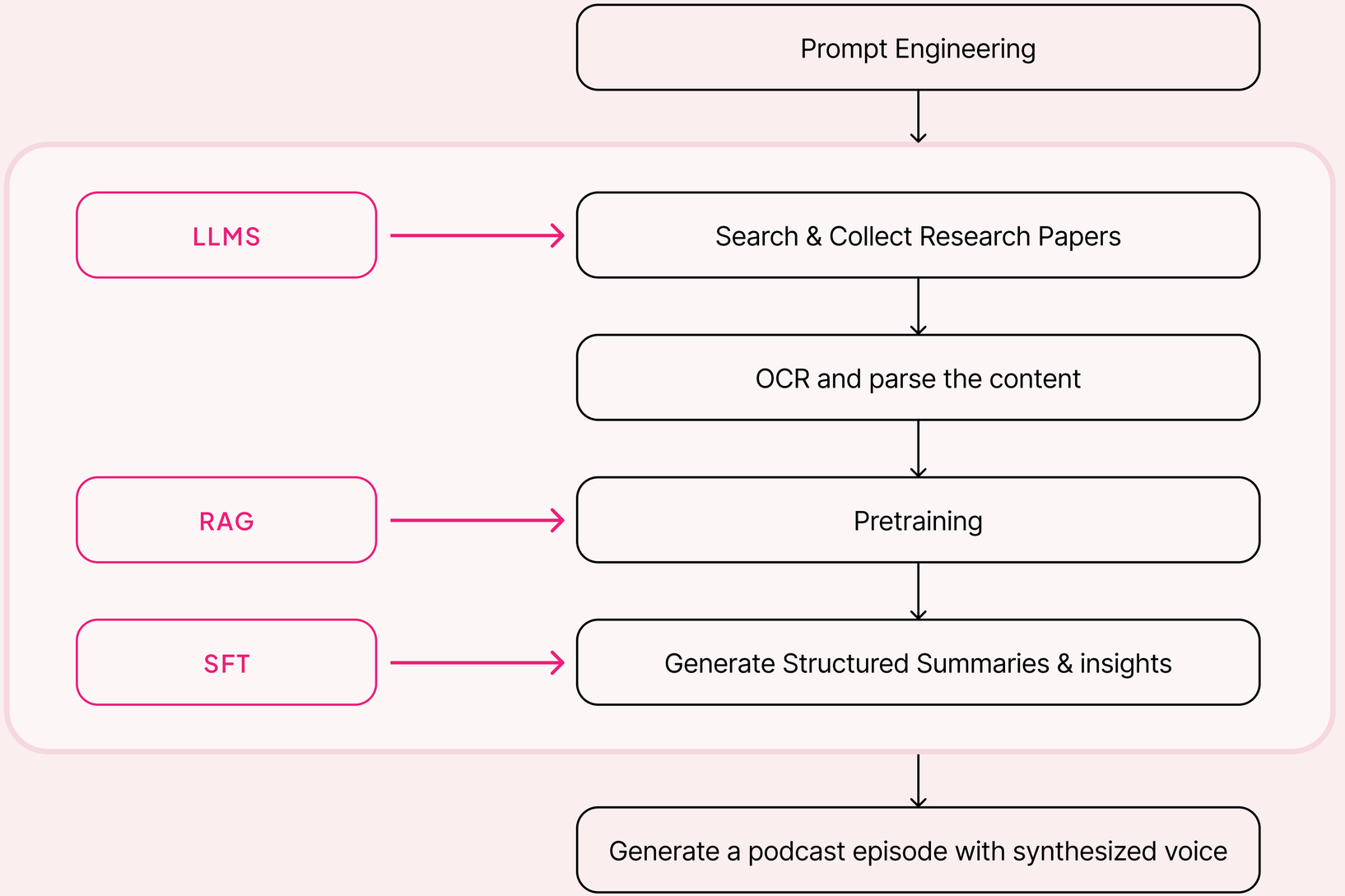
Course Schedule

Weeks 8–10: Course Project & Wrap-up

Week	Topic	Lecture Themes	Project
9	Voice Agent (Multimodal AI, GPT-4o, ASR/TTS)	GPT-4o real-time, Emilia pipeline, chained vs end-to-end agents	Build a voice agent (GPT-4o style), optionally explore NotebookLM-like pipeline
10	Final Capstone Project: Research Agent	Agents, MCP protocol, function calling, task chaining	End-to-end pipeline: search papers → OCR → summarize → generate podcast → voice agent output

What We're Building

Research Agent: The Final Project



Generative AI

Systems that generate new content (text, images, audio, code)

Agentic AI

Autonomous task-completing systems that use generative models

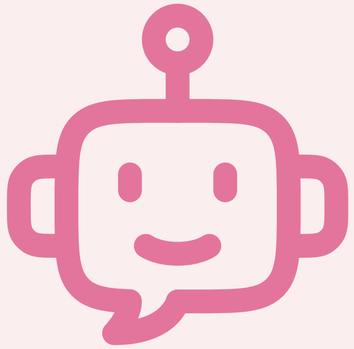
LLMs

LLMs are the core of most GENAI system Today

Large Language Models (LLMs)

- Trained on massive internet-scale text corpora
- Predict the next token based on prior context
- Capable of reasoning, summarizing, translating, coding, and more

LLM Applications



To-Consumer

Chatbots, Virtual Tutors,
Personal Assistants, Copilots

To-Enterprise:

Contract Analysis, Research
Automation, Customer
Service, Code Review

The Lifecycle of an LLM

Pretraining → SFT → Alignment

Use next-token prediction during pretraining

Scaling laws guide model size & data size

How We Use LLMs

Prompt

THE MESSAGE WE SEND TO THE LLM

System Prompt

CONFIGURES BEHAVIOR

User Prompt

TASK-SPECIFIC INPUT

Types

ZERO-SHOT,FEW-SHOT, CHAIN-OF-THOUGHT, AGENT

Best Practices with CO-STAR

Context

Provide background

Objective

Define clear goals

Style

Formal vs casual, concise vs verbose

Tone

Authoritative, exploratory...

Audience

Who's reading the result?

Response Format

Define output structure (e.g.JSON)

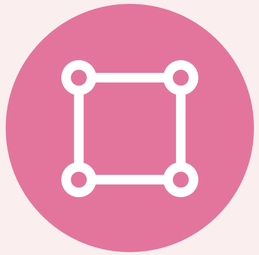
Practice: Prompting for Research Agents

Design > **Rewrite** > **Explore** > **Identify**

Design	Rewrite	Explore	Identify
Design a prompt to act as your research agent	Rewrite it using CO-STAR	Explore structured outputs (JSON, XML)	Identify model limitations and edge cases

What is MCP? Why Should We Care?

- The Model Context Protocol (MCP) is an open protocol that enables seamless integration between LLM applications and external data sources and tools.
- MCP enables Claude Desktop to interact seamlessly with external tools and AI models, enhancing its capabilities.



MCP = LIGHTWEIGHT, MODULAR PROTOCOL FOR CHAINING AI TOOLS

ENABLES INTEROPERABILITY BETWEEN TOOLS, APLS, AND MODELS

CORE TO BUILDING AGENTS WITH MULTIPLE CAPABILITIES (SEARCH + OCR+SUMMARY)

MCP Setup Instructions

Prerequisites

1. Claude Desktop Application: Ensure it's installed and updated to the latest version.
2. Node.js: Required for running MCP servers.

- Download: Visit Node.js Official Website and download the LTS version.
- Installation: Run the installer and follow the on-screen instructions.

Step-by-Step MCP Integration

1. **Locate the Configuration File:**

- Open Claude Desktop and go:
- Setting – developers – edit config

2. **Edit the Configuration File**

- Open the claude_desktop_config.json file in a text editor.
- Add or update the mcpServers section with the desired server configuration.

3. **Restart Claude Desktop**

MCP Server Example

Online function: Brave-search

An MCP server implementation that integrates the Brave Search API, providing both web and local search capabilities.

```
{
    "mcpServers": {
    "brave-search": {
        "command": "npx",
        "args": [
        "-y",
        "@modelcontextprotocol/server-brave-search"
        ],
        "env": {
        "BRAVE_API_KEY": "YOUR_BRAVE_API"
        }
    }
    }
}
```

MCP Server Example

File access: server-filesystem

Node.js server implementing Model Context Protocol (MCP) for filesystem operations.

```
{
    "mcpServers": {
        "filesystem": {
            "command": "npx",
            "args": [
              "-y",
              "@modelcontextprotocol/server-filesystem",
              "/Users/username/Desktop"
            ]
        }
    }
}
```

MCP Server Example

Sequential Thinking: server-sequential-thinking

An MCP server implementation that provides a tool for dynamic and reflective problem-solving through a structured thinking process.

```
{
    "mcpServers": {
        "sequential-thinking": {
            "command": "npx",
            "args": [
              "-y",
              "@modelcontextprotocol/server-sequential-thinking"
            ]
        }
    }
}
```

Project Schedule

Week	Milestone	Description
1	Project Kickoff	Introduction to course project: "Your Personalized Research Agent". Define goals & start experimenting with prompts.
4	Project Insight I (TA Section)	First review of project idea. Share initial progress, receive peer/TA feedback. Adjust scope if needed.
7	Project Insight II (TA Section)	Second review checkpoint. Final decision: lock in project direction & components.
10	Final Presentation	Present your working agent: share learnings, showcase demo, and reflect on technical depth.

Kickstart of Your Project

- **Explore Use Cases**
  Investigate real-world research agents (e.g., paper summarizers, citation tools, QA systems). Brainstorm how LLMs can boost your own research workflows.
- **Define Your Agent's Goal**
  Write a 1-sentence mission:
  "My agent helps me [task] by [method]."
- **Start Prompt Engineering**
  Design your first prompt using CO-STAR and chain-of-thought techniques.
  Format output with JSON/XML for structured results.
- **Join Discord & Collaborate**
  Introduce yourself. Share your agent idea. Join a feedback group.
- **Study Examples**
  Review prompt demos & agent walkthroughs. Note what works — and what doesn't.
- **Track Model Limitations**
  Record hallucinations, logic gaps, or failures — these will shape your future projects (RAG, SFT, alignment).

Key Takeaways

- Generative AI is powerful — but needs smart prompting
- LLMs respond differently based on how we talk to them
- Prompt engineering is both art and science
- MCP is our agent-building protocol — learn it well!

Homework

- Complete all tasks in Project 1 (see class repo)
- Join Discord group for Q&A and code sharing
- Setup MCP server in your local
- Write a 1-sentence project idea and share in the Discord group
- Next week: deep dive into Transformers and Pretraining

Thank you!