## Fine-Tuning Large Language Models (LLMs) Week-5: Non-Technical Assignment

### Agentic AI and Fine-tuning from Lecture 1 (8 points) Rag and Agentic-RAG (7 Points)

#### Section 1: Understanding LLM-Based Agents ( 0.5 Point)

##### 1: How do AI agents work autonomously? (Answer concisely in 2-4 lines.)¨

AI agents work autonomously by processing inputs, reasoning based on predefined models or learned patterns, and generating actions without human intervention. They utilize techniques like reinforcement learning, prompt engineering, and memory to adapt to tasks dynamically. Large Language Model (LLM)-based agents specifically rely on deep learning architectures to understand context and generate human-like responses.

#### Section 2: Test Case Scenario Generation (1 Point)

##### 2. Scenario:

In class, you observed how user stories are created and prioritized using AI agents. Now, your focus is on:

* Generating test case scenarios for the user stories.

##### 3. Your Task:

* Select a minimum of 2 agents (you may include more as needed).
* List the agents you’ve chosen and roles.
* Provide sample prompts that each agent would use to generate test case scenarios.

##### 4. Deliverables:

|  |  |  |
| --- | --- | --- |
| Agent Name | Role | Prompt |
| Product Owner Agent | The PO Agent is responsible for generating user stories based on project descriptions provided by stakeholders. | The PO Agent is responsible for generating user stories based on project descriptions provided by stakeholders. It ensures that user stories capture customer requirements clearly and concisely. The agent analyzes project details and formulates structured user stories in the format:  "As a [user], I want [goal] so that [benefit]." |
| Quality Assurance Agent | The QA Agent evaluates user stories using the INVEST framework (Independent, Negotiable, Valuable, Estimable, Small, and Testable). It ensures that each story is clear, self-contained, and actionable. | The QA Agent evaluates user stories using the INVEST framework (Independent, Negotiable, Valuable, Estimable, Small, and Testable). It ensures that each story is clear, self-contained, and actionable. The agent checks for ambiguities, verifies the presence of acceptance criteria, and refines stories to align with best practices. |

### Section 3: Multi-Agent Fine-Tuning and Research References (1 Point)

#### 3. Can multi-agent-based fine-tuning be implemented? If yes

* List the key steps for implementation.
* Mention at least one research paper or reference that supports this. (Keep it concise.)
* IF NO, give a reason.

Yes, it can be implemented

1. **Start with a Pre-Trained Base Model**
   1. We take an existing LLM (e.g., GPT-4, Mistral, LLaMA-3).
   2. Duplicate it to create **multiple independent copies**.
2. **Fine-Tune the Model into Specialized Agents**
   1. **Generation Agents:** Fine-tuned to generate responses based on queries.
   2. **Critic Agents:** Fine-tuned to evaluate and refine responses from generation agents.
3. **Multi-Agent Debate & Iterative Improvement**
   1. Generation agents generate multiple responses.
   2. Critic agents analyze and provide feedback.
   3. The system **votes** on the best response and uses it for further fine-tuning.
4. **Repeat Fine-Tuning for Multiple Iterations**
   1. Each iteration fine-tunes the agents further using new datasets collected during interactions.
   2. This process **diverges** the models, making each agent more specialized based on their generations in the previous rounds of finetuning

**Final Result: Many Fine-Tuned Models**

Source: Subramaniam et al. (2025), "Multiagent Finetuning: Self Improvement with Diverse Reasoning Chains", <https://arxiv.org/pdf/2501.05707>, Project website at <https://llm-multiagent-ft.github.io/>

### Section 4: Techniques for Fine-Tuning Beyond PEFT & LoRA (0.5 Point)

#### 4. What are other fine-tuning techniques besides PEFT, LoRA, Q-LoRA, and Prompt Fine-Tuning and full fine-tuning?

* **List a few of the names.** (Include at least one paper reference.) (Keep it concise.

Here are some alternative fine-tuning techniques beyond PEFT, LoRA, Q-LoRA, Prompt Fine-Tuning, and Full Fine-Tuning:

1. Self-Rewarding Fine-Tuning (SRFT)

* Uses reinforcement learning where a model self-evaluates and fine-tunes based on its own generated feedback.
* Paper: *Yuan et al. (2024), "Self-Rewarding Language Models"* ([arXiv:2401.10020](https://arxiv.org/abs/2401.10020)).

2. Direct Preference Optimization (DPO)

* Fine-tunes LLMs by optimizing responses based on human preferences without requiring reinforcement learning.
* Paper: *Rafailov et al. (2024), "Direct Preference Optimization: Your Language Model is Secretly a Reward Model"* ([arXiv:2305.18290](https://arxiv.org/abs/2305.18290)).

3. Iterative Reasoning Preference Optimization (IRPO)

* Enhances reasoning capabilities by iteratively fine-tuning on preferred reasoning paths.
* Paper: *Pang et al. (2024), "Iterative Reasoning Preference Optimization"* ([arXiv:2404.19733](https://arxiv.org/abs/2404.19733)).

4. Distillation-Based Fine-Tuning

* Uses a smaller student model that learns from a larger model's outputs.
* Paper: *Hsieh et al. (2023), "Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes"* ([arXiv:2305.02301](https://arxiv.org/abs/2305.02301)).

5. Multi-Agent Fine-Tuning (MAFT)

* Fine-tunes multiple specialized agents that interact through debate and iterative learning.
* Paper: *Subramaniam et al. (2025), "Multiagent Finetuning: Self-Improvement with Diverse Reasoning Chains"* ([arXiv:2501.05707](https://arxiv.org/abs/2501.05707)).

### Section 5: Designing an AI-Powered Fine-Tuning Application for Next generation User Needs (5 Points)

Scenario:

*As a CEO or entrepreneur with a vision to develop an agent-based AI-powered application that empowers non-technical users to easily fine-tune specialized AI models (e.g., LLMs, vision models, etc.) tailored to their unique business needs, you are positioned at the forefront of the multi-agent development wave in 2025. Your goal is to enable users to create custom AI agents or fine-tuned models that seamlessly integrate into their services or products without requiring deep technical knowledge. However, you face challenges, including managing a small but highly skilled team with limited resources and expertise in agent-based application development, alongside a tight development timeline that demands careful prioritization of features to ensure timely delivery and business growth.*

An important system is under development. You may encounter issues. If user stories are not generated by the system, you can use AI to create them yourself, then upload them to the system and prioritize them.

Key Features of the Application: NOTE (can be added more)

* Multi-Modal Interaction: Users can describe their needs via chat, voice, or calls.
* Real-Time Dashboard: Provides instant feedback, tracks fine-tuning progress, displays performance metrics, and offers insights into cost and time estimations.
* Automated Workflow: AI agents handle the entire process, including:
  + Data Processing
  + Model Selection
  + Fine-Tuning
  + Evaluation & Testing
  + Deployment into User Products or Services
* Specialized Agent Creation: The system delivers a specialized AI model or agent that users can embed into their existing software ecosystems or business workflows.
* Performance Monitoring: Ongoing tracking of the model’s real-world performance with optimization suggestions.

##### Answer

Separate .xlsx file

### Section 6: AI vs. Human CEOs

A Fortune 500 company is exploring the possibility of replacing its CEO with an AI-CEO. This AI-CEO would be responsible for making all strategic decisions, using real-time corporate data, economic forecasts, and industry best practices to guide the company's direction.

* Question: Your task is to develop a business case arguing for or against the adoption of an AI-CEO. The goal is to provide a thoughtful, evidence-based perspective on whether an AI-CEO would be a better choice for an organization.  
  Your business case should address the following key areas:
  + How can an AI-CEO using RAG understand and use real-time business information to make better decisions than a human CEO?
    - It would be a poor decision to replace the CEO of a Fortune 500 company with an AI-CEO. A Fortune 500 company is so large that the decisions a CEO must make are not routine; they are unique and case specific. There are no existing databases for RAG (Retrieval-Augmented Generation) that contain such specialized decision-making information. How about black swan events?  
      Additionally, a human CEO has the ability to motivate and inspire people, driving them to perform better. Can AI truly replicate that? Moreover, company owners would likely be reluctant to grant full decision-making control to AI.
  + What vector databases and embedding models can be used to retrieve relevant industry trends?
    - Query a vector database with real-time financial reports, news articles, and research papers. Good vector databases are for example Pinecone, Weaviate or FAISS.
    - To extract meaningful insight embedding models like ada-002, SBERT or USE can be used.
  + What orchestration frameworks (e.g., LangChain, LlamaIndex) can be used to build an AI-CEO with autonomous decision-making capabilities?  
    Here are some:
    - **LangChain**
      * Best for: Complex decision-making workflows, integrating multiple AI tools and databases.
      * Capabilities:
        + Chains LLMs with external data sources (APIs, databases, reports).
        + Manages reasoning steps using agent-based execution.
        + Supports memory for long-term context awareness.
        + Works well with vector databases (e.g., Pinecone, Weaviate).
      * Use Case:
        + Automating financial trend analysis.
        + Building multi-step decision processes for business strategies.
    - **LlamaIndex (Formerly GPT Index)**
      * Best for: Organizing and retrieving structured business intelligence data.
      * Capabilities:
        + Connects LLMs with private business documents, research reports, and industry data.
        + Uses structured indices for better RAG (Retrieval-Augmented Generation) performance.
        + Enhances AI-CEO's ability to query internal company knowledge bases.
      * Use Case:
        + AI-CEO accessing past business decisions for reference.
        + Legal and compliance decision-making automation.
    - **AutoGen**
      * Best for: Multi-agent AI architectures with self-improving decision-making loops.
      * Capabilities:
        + Allows multiple AI agents to collaborate (e.g., CEO-Agent, Market-Research-Agent, Finance-Agent).
        + Supports autonomous reasoning and self-improving logic.
        + Ideal for AI-driven strategic planning and risk management.
      * Use Case:
        + AI-CEO evaluating investment risks with multiple autonomous agents.
    - **CrewAI**
      * Best for: AI-powered executive teams working together to optimize decisions.
      * Capabilities:
        + Enables hierarchical AI agents with specialized roles (CEO, CFO, Analyst).
        + Improves long-term business planning via agent cooperation.
      * Works with LangChain and LlamaIndex for decision-making workflows.
      * Use Case:
      * AI-CEO working alongside an AI-Financial Officer and AI-Strategy Officer.
    - **Flowise AI**
      * Best for: No-code/low-code orchestration of LLM-based decision-making pipelines.
      * Capabilities:
        + Provides visual workflow design for AI-CEO decision flows.
        + Supports API and third-party service integration for business data integration.
        + Works well for non-technical implementations of AI-driven business automation.
      * Use Case:
        + Automating AI-CEO decision workflows without complex programming.
  + What challenges does an AI-CEO using RAG face when dealing with ethics, laws, and crises compared to a human CEO?
    - Ethical Challenges
      * No Moral Judgment – AI lacks human intuition for complex ethical trade-offs.
      * Bias Risks – AI inherits biases from training data, leading to unfair decisions.
      * CSR Limitations – May prioritize profit over long-term social responsibility.
    - Legal Challenges
      * Regulatory Ambiguity – Struggles with interpreting evolving laws.
      * Liability Issues – AI cannot be legally accountable; board members remain responsible.
      * Data Privacy Risks – May mishandle sensitive data under laws like GDPR.
    - Crisis Management Challenges
      * No Emotional Intelligence – Cannot reassure employees or investors in crises.
      * Rigid Decision-Making – Lacks human adaptability in uncertain situations (black swan events)
      * Risk Aversion – May avoid bold moves that require intuition.
  + How do Agentic RAG models ensure coordination between AI Agents managing different business domains?
    - Hierarchical Agent Structure – Assigns roles (e.g., Strategy-Agent, Finance-Agent, HR-Agent) with clear responsibilities.  
      Shared Memory & Context – Agents access a common knowledge base to align on business goals.  
      Cross-Agent Communication – Uses message passing and API calls to share insights and decisions.  
      Consensus Mechanisms – Implements voting, reinforcement learning, or weighted prioritization for final decisions.  
      Feedback Loops – Continuously refines outputs based on real-time data and human oversight.
  + How to get real time data for RAG (inside and outside of the company)
    - APIs & Web Scraping – Pulls data from financial, news, and market sources (e.g., Alpha Vantage, Google News).
    - Database Integration – Connects to real-time databases (SQL, NoSQL, vector DBs like Pinecone).
    - Streaming Data Pipelines – Uses Kafka, Pub/Sub, or WebSockets for continuous updates.
    - IoT & Sensor Data – Integrates with business systems for live operational data.
    - Human Feedback Loops – Captures expert input to refine real-time responses.
  + How AI-CEO would communicate to company workers?
    - Automated Reports – Sends data-driven insights via emails, dashboards, and notifications.
    - AI Chat Assistants – Provides real-time responses to employee inquiries.
    - Virtual Meetings – Delivers video or text briefings through internal platforms.
    - Task Automation – Assigns and tracks work through project management tools.
    - Sentiment Analysis – Monitors employee feedback and adjusts communication style.

You are encouraged to apply real-world examples of AI-driven decision-making from companies such as Amazon, Tesla, and Google.

### Section 7: AI-Powered Work Trip Planner

Your company wants to develop an AI-powered work trip planner that helps employees check their travel eligibility, suggest flights and hotels, and automate expense approvals. It will provide real-time travel updates, such as flight availability, hotel pricing, and weather conditions. Your task is to explain how vector databases and LLMs can be used to solve these challenges.  
You can watch this video for understanding: [What are AI Agents](https://www.youtube.com/watch?v=F8NKVhkZZWI)

Question:

* What data should be stored as vector embeddings and how can a vector database and LLMs work together to retrieve real-time travel data (e.g., flights, hotels, weather)?
* Should the system use a Retrieval-Augmented Generation (RAG) approach or external API calls? Justify your choice. What advantages does a chat-based LLM tool bring to a work trip planner, and what risks should be considered when deploying such a solution?

Deliverables: Write answers in two paragraphs for the given questions.

The system should store employee preferences, company travel policies, past booking history, frequent destinations, and personalized travel constraints as vector embeddings. These embeddings allow the AI to retrieve the most relevant travel options quickly. A vector database (e.g., Pinecone, Weaviate, FAISS) enables similarity searches by matching user queries with stored embeddings, retrieving travel options tailored to the user’s preferences. LLMs process these retrieved results, refine responses, and provide structured recommendations for flights, hotels, and expenses. Additionally, real-time travel data (e.g., flight schedules, hotel prices, and weather conditions) is fetched via external APIs and integrated into the response pipeline, ensuring accuracy and up-to-date recommendations.

A Retrieval-Augmented Generation (RAG) approach is preferred over purely external API calls because it allows for context-aware responses by combining internal knowledge (company policies, past trips) with real-time travel data. However, API calls remain essential for live updates. A chat-based LLM tool improves user experience by enabling natural interaction, allowing employees to ask travel-related questions, adjust bookings, and receive personalized itinerary updates without navigating complex interfaces. Risks include data privacy concerns, potential hallucinations in AI-generated responses, and delays in integrating real-time travel data, requiring human oversight mechanisms and API validation layers to maintain accuracy and compliance.

### Section 8: Draw a Sequence Diagram for a RAG-Based Medical Consultation System

* Question: Imagine a teleconsultation platform where a Patient seeks medical advice from an online system. Under the hood, the system implements RAG for context-rich answers. In addition, it can perform iterative/agentic retrieval if more information is needed, essentially RAG with the option to refine queries before finalizing a response.
  + Patient: The individual describing symptoms and asking questions.
  + Teleconsultation App: The front-end interface that receives the patient’s input and displays the system’s response.
  + RAG Engine: Combines Retrieval and Generation in one module. Performs the initial RAG steps (fetch relevant data, generate an answer). Can also perform “agentic” or iterative retrieval if it detects missing info or ambiguities.
  + EHR System (Electronic Health Records): Stores the patient’s medical history, previous diagnoses, lab results, etc.
  + Medical Knowledge Base: A repository of medical guidelines, research articles, or other reference materials that might be relevant to the patient’s condition.

Deliverables: Sequence Diagram and Short Explanation  
  
The Self-RAG Based Medical Consultation System is an AI-powered teleconsultation platform designed to provide accurate and context-rich medical advice. The system begins when a patient submits symptoms and medical queries through a teleconsultation app. Once received, the RAG Engine processes the query by retrieving relevant information from two primary sources. The patient’s medical history is accessed through an API-based Electronic Health Records (EHR) system, while additional medical guidelines and references are retrieved from a vectorized Medical Knowledge Base using semantic search.

After gathering the necessary information, the system generates an initial response using a Large Language Model (LLM). To ensure accuracy and completeness, a self-reflection mechanism evaluates whether the response is sufficient. If gaps or ambiguities are detected, the system refines the query and reinitiates the retrieval process, searching for additional context before generating an improved response. This iterative refinement continues until a satisfactory answer is obtained.

Once the response is finalized, it is delivered back to the teleconsultation app, where it is displayed to the patient. By combining personalized medical history with advanced retrieval techniques, the Self-RAG system enhances AI-driven telemedicine, ensuring that the responses are both relevant and reliable while reducing the risk of misinformation or hallucinations.

A diagram of a medical procedure

AI-generated content may be incorrect.