Generative AI (Spring-2025)

Group Assignment - 4

Instructor

Dr. Hajra Waheed PHD

Submission Guidelines:

- This is a **group assignment** of up to **4 people per group**.
- Only 1 Student need to submit per group
- Mention names and roll numbers in the report.
- Submit your assignment on Google Classroom in the format "20XX.zip".
- The deadline is May 6, 2025, at 11:59 PM. No extensions will be granted.

Declarations:

- No Late submissions will be accepted.
- Plagiarism will result in zero marks for the assignment.
- Please ensure that you submit your own original work.

VIVA Policy:

- A VIVA (oral examination) will be conducted to assess your understanding of the assignment.
- The VIVA will be scheduled separately, and you will be notified of the date and time.
- Failure to attend the VIVA will result in zero marks for the assignment.

Academic Integrity:

- Plagiarism, collusion, and academic dishonesty will not be tolerated.
- Any instances of academic misconduct will be reported to the authorities and may result in severe penalties.

Smart Summarizer

Fine-Tuning LLMs with LoRA to Understand Academic Research Papers

Objective

Reading academic papers is time-consuming and mentally demanding, particularly for undergraduates, researchers, and professionals who require quick and accurate insights. This assignment aims to develop an intelligent summarization system by fine-tuning a Large Language Model (LLM) using LoRA (Low-Rank Adaptation) — a parameter-efficient fine-tuning technique that introduces a small number of trainable weights, reducing computational cost while retaining effectiveness. The resulting system should generate accurate and readable summaries of academic texts, thereby making the research review process significantly more efficient.

In addition to traditional automatic evaluation metrics such as ROUGE, BLEU, and BERTScore, this assignment incorporates **LLM-as-a-Judge**, a modern evaluation framework where a powerful, general-purpose LLM is prompted to assess the quality of generated summaries. These evaluations are based on three human-aligned criteria: **fluency**, **factuality**, and **coverage**. This allows for a more nuanced, qualitative judgment of summary performance, reflecting how end-users might perceive the output. By combining both statistical and human-aligned evaluations, the assignment provides a comprehensive assessment of the summarization system's effectiveness

Overview

Component	**Description**
Dataset	arXiv summarization dataset (abstracts + full papers) https://huggingface.co/datasets/ccdv/arxiv-summarization
Model	Pre-trained LLM (e.g.**, LLaMA 3, Mistral 7B) with LoRA fine-tuning**

Training Goal	Summarize research papers accurately using minimal trainable parameters
Evaluation	ROUGE, BLEU, BERTScore, and **LLM-as-a-Judge** for qualitative evaluation
App	Streamlit or Gradio-based interface for interactive summarization testing
Agent	Multi Agent system

Assignment Tasks

Part 1: Data Preprocessing

- Load the arXiv summarization dataset.
- Select a subset of **5,000 samples**.
- Extract input and target pairs:
    - **Input**: Article.
    - **Target**: Abstract/Summary
- Tokenize the dataset using the tokenizer from your base model.
- Split the data into **Training (80%)**, **Validation (10%)**, and **Test (10%)** sets.

Part 2: LoRA-Based Fine-Tuning

- Select a suitable pre-trained model such as:
    - llama-3-7b
    - mistral-7b
- Integrate **LoRA** using the HuggingFace PEFT library.
- Configuration Parameters:
    - r = 8
    - alpha = 16
    - dropout = 0.1
    - Apply LoRA to attention layers q and v.
- Train the model for **4 to 5 epochs**.
- Save the trained model.

Part 3: Inference and Output

- Use your fine-tuned model to generate summaries on **10 unseen samples** from the test set.
- Additionally, generate:
    - Summaries using the **base (non-fine tuned) model**.
    - Ground-truth Abstracts/**summaries** from the dataset.
- Save and compare all outputs.

Part 4: Model Evaluation

A. Automatic Evaluation (Quantitative)

For each summary, calculate the following metrics:

Metric	Description	Tool/Library
ROUGE-1	Measures word-level overlap	rouge_score
ROUGE-L	Measures longest common subsequence	evaluate or rouge_score
BLEU	Measures phrase-level overlap (n-gram precision)	nltk.translate.bleu_score
BERTScore	Measures semantic similarity using embeddings	bert_score (HuggingFace)

- Visualize all evaluation scores using **bar charts or tables**.
- Discuss the performance comparison between the fine-tuned model and the base model.

B. LLM-as-a-Judge Evaluation (Qualitative)

In this section, use a powerful LLM to serve as an **automated evaluator** of summaries. You may use prompting or chain-of-thought to guide the LLM in assessing each summary based on three key dimensions:

Judges you can explore on https://www.together.ai/ you will get free 1$ credit, use wisely

1. **Meta Llama 3.1 70B Instruct Turbo**
2. **DeepSeek V3-0324/**
3. **meta-llama/Llama-4-Maverick-17B-128E-Instruct-FP8**

Criterion	Description
Fluency	Is the summary readable and grammatically correct?
Factuality	Are the statements in the summary correct, and do they reflect the source?
Coverage	Does the summary include the main problem, method, and key findings?

Instructions:

- Use 10 randomly selected summaries.
- For each summary, prompt the LLM to rate each dimension on a **scale of 1 to 5**.
- Calculate the average score for each dimension.
- You will use:
    - An automated script using Together.ai API

Sample Prompt Format for Evaluation <u>build your own better version</u>

```
Given the following input and the summary produced, evaluate the summary on

1. Fluency

2. Factuality

3. Coverage

Use a score from 1 (poor) to 5 (excellent) for each. Provide a short justification for
each score.

Input: [Original paper text]

Generated Summary: [Your Model's Summary]
```

Part 5: App

Develop a minimal **Streamlit or Gradio-based interface** that allows users to:

- Upload a research paper (PDF or plain text).
- Click the “Summarize” button.
- Display the generated summary with options to compare it with the base model.
- LLM-as-a-judge also score this generated Summary

Part B: Multi-Agent Autonomous Research Assistant using LangGraph and LLMs

Objective

The objective is to design and implement a multi-agent autonomous research system using **LangChain**/**LangGraph**. This system will automate the academic research process by:

- Accepting user-provided research keywords
- Retrieving relevant academic papers from multiple sources
- Ranking and selecting the most significant papers
- Summarizing their contents using a your above fine-tuned **Model**
- Comparing insights to identify common themes, contradictions, and research gaps

This system aims to reduce the manual effort required in literature review and streamline the identification of high-impact academic content.

🤖 Agents

The system consists of five specialized agents orchestrated using **LangGraph**:

1. **KeywordAgent**
   Enhances the user’s input by generating expanded and related keywords using an LLM, improving the accuracy and coverage of search queries.
2. **SearchAgent**
   Interfaces with academic search APIs (e.g., arXiv, Semantic Scholar, PubMed) to retrieve relevant papers based on the expanded keywords.
3. **RankAgent**
   Scores and ranks papers using a multi-criteria strategy including:
   - Citation count

    - Publication date
    - Relevance to keywords (inferred through an external LLM API such as Together.ai)
4. **SummaryAgent**
   Processes selected top-ranked papers and generates structured summaries using a **LoRA fine-tuned language model** which you already trained for academic summarization.
5. **CompareAgent**
   Performs comparative analysis on the summarized content to:
    - Identify consensus and disagreements
    - Highlight underexplored areas or research gaps
    - Suggest possible directions for future work
    - **Hint Use LLM**

Tasks

The autonomous agent system performs the following tasks:

1. **Keyword Expansion**
   Parses and enhances user input into multiple meaningful search terms for broader paper retrieval.
2. **Literature Search**
   Conducts academic searches across multiple repositories using APIs, retrieving metadata such as title, abstract, authors, year, and citations.
3. **Paper Evaluation and Ranking**
   Filters and ranks papers using pre-defined metrics and LLM-inferred relevance to identify 3–5 most impactful publications.
4. **Summarization**
   Extracts key content (e.g., abstract, introduction, methods, conclusion) from selected papers and generates clear summaries using a fine-tuned model.
5. **Comparative Analysis and Insight Extraction**
   Compares the summarized papers to determine shared findings, conflicting results, and areas that need further research attention.

Final Output

The system generates a research report consisting of:

- **Topic Summary**
  A brief overview of the topic as expanded and interpreted by the agent.
- **Top Papers List**
  A ranked list of selected papers with the following details for each:
    - Title
    - Authors
    - Full paper Summary (fine tuned Model)
    - Methodology

    - Key contributions
    - Limitations/Gaps
- **Comparative Analysis**
  A section identifying:
    - Common findings across papers
    - Contradictory or conflicting insights
    - Research gaps and underexplored areas

This output can be exported in structured formats such as PDF or visualized or integrated into a Streamlit-based UI.

Helping Materiels:
http://export.arxiv.org/api/query?search_query=all:LLM&start=0&max_results=10

Must Watch
https://www.youtube.com/watch?v=wSxZ7yFbbas

https://www.youtube.com/watch?v=hvAPnpSfSGo

https://www.youtube.com/watch?v=Q866J2m8hxs

https://www.youtube.com/watch?v=rCWpJlZdH0c

Final Report

Prepare a well-structured and professional PDF report, containing:

1. **Dataset Overview**
    - Dataset source, preprocessing details, tokenization
2. **Model and LoRA Configuration**
    - Architecture used, hyperparameters, LoRA setup
3. **Training Logs & Observations**
    - Training/validation loss curves, runtime environment and GPU usage.
4. **Output Samples**
    - Table comparing generated summaries: base model, LoRA, human
5. **Evaluation Results**
    - Quantitative: ROUGE, BLEU, BERTScore (with visualizations)
    - Qualitative: LLM-as-a-Judge prompts and ratings
6. **Agent Structures and prompts you have used.**

Submission Checklist ZIP

- Code notebook (training + inference outputs) evaluation results of fine tuning
- Your Fine Tuned model app so we can test Summarizer independently.
- Your working Agent.
- Final report (PDF)