***Legal Document Summarization via LoRA-Fine-Tuned LLM***

**1. Project Objectives**

* **Primary Goal:** Apply a LoRA-based fine-tuning of a pre-trained LLM (Qwen2.5-1.5B-Instruct) to generate concise summaries of long-form legal documents.
* **Secondary Goals:**
  + Compare performance of the base model vs. the LoRA-fine-tuned model.
  + Produce structured, reliable JSON outputs (Summary\_long, Summary\_short).
  + Demonstrate reproducible, end-to-end fine-tuning workflow.

**2. Project Discussion**

**2.1 Project Idea**

A clear, concise description of your task:

We aim to fine-tune a large language model (Qwen2.5-1.5B-Instruct) via Low-Rank Adaptation (LoRA) to automatically summarize lengthy legal case documents into both long-form and short-form summaries, outputting results in JSON for downstream processing.

**2.2 Dataset Information**

* **Source:** Multi-LexSum dataset (legal case summaries).
* **Size & Splits:**
  + Training: 3000 documents
  + Validation: 400 documents
  + Test: 400 documents
* **Structure:** Each record contains:
  + sources: list of text passages
  + summary/long: human-written long summary
  + summary/short: human-written short summary
* **Preprocessing Steps:**
  + Loaded dataset from disk with datasets.load\_from\_disk.
  + Truncated or concatenated up to three source passages.
  + Packaged into chat-style JSON examples for fine-tuning.

**2.3 Base Model Information**

* **Model Name:** Qwen/Qwen2.5-1.5B-Instruct
* **Architecture:** 2.5 B-parameter causal LM with instruction-tuning capability.
* **Capabilities:** Strong general-purpose instruction following; supports long context (up to 3,500 tokens).

**2.4 Brief Explanation of Fine-Tuning with LoRA**

* **What is LoRA?** Low-Rank Adaptation injects trainable low-rank matrices into each transformer layer, drastically reducing parameter update cost while preserving pre-trained weights.
* **How Applied:**
  1. Set finetuning\_type: lora, lora\_rank: 64, lora\_target: all.
  2. Configured training in summarizer\_finetune.yaml.
  3. Conducted two epochs of fine-tuning with bf16 precision on NVIDIA L4 GPUs.

**2.5 Evaluation Methods**

* **Pre- vs. Post-Fine-Tuning Comparison:**
  + **BLEU** (focus on 1–4-grams with custom weighting)
  + **ROUGE** (1, 2, and L)
  + **BERTScore** (precision, recall, F1)
* **Sampling:** Evaluated on 10 test samples for rapid iteration; full test planned for final results.
* **Tools:** NLTK, Hugging Face’s rouge\_scorer, bert\_score.

**2.6 Project Results and Plots**

* **Quantitative Results (10-sample subset):**

|  |  |  |
| --- | --- | --- |
| **Metric** | **Base Model** | **LoRA-Fine-Tuned** |
| BLEU | 0.0000 | 0.0979 |
| ROUGE-1 F1 | 0.0136 | 0.3827 |
| ROUGE-2 F1 | 0.0068 | 0.1740 |
| ROUGE-L F1 | 0.0136 | 0.2061 |
| BERTScore F1 | 0.7952 | 0.7773 |

* **Visual Plots:**
  + Loss curves over training steps (generated by LoRA-Factory’s plot\_loss: true).
  + Bar chart comparing ROUGE-1/2/L before vs. after (can be added via Matplotlib).

2.7 Structured Output Examples

{

"input\_document": "…first three source passages…",

"Summary\_long": "Detailed summary of the legal case…",

"Summary\_short": "Concise one-sentence summary…"

A screenshot of a graph

AI-generated content may be incorrect.}

A screenshot of a graph

AI-generated content may be incorrect.