LLM Fine-Tuning for Email Summarization And Response

Prachi Naidu
pnaidu2@gmu.edu

Riya Elizabet Stanly
rstanly@gmu.edu

Archana Haridas
aharida@gmu.edu

1 Recap: Brief Project Description

Our project focuses on building a lightweight, instruction-tuned Large Language Model (LLM) designed for automated email summarization and context-aware reply generation. Email threads often contain noise such as signatures, forwarded headers, and quoted text, making them lengthy and difficult to process efficiently. Our goal is to reduce this cognitive load by creating a system that intelligently compresses the content and generates professional replies tailored to user instructions.

To achieve this, we developed an integrated model that accepts raw email threads as input, cleans the content through preprocessing, and produces both a concise summary capturing essential information and a coherent, contextually appropriate reply. Using LoRA fine-tuning on a small open-source LLM (meta-llama/Llama-3.2-1B-Instruct), we created a system that provides strong semantic performance while remaining computationally efficient and accessible for academic experimentation.

2 Detailed System or Methodology Description

In this project, we developed an instruction-tuned large language model designed to perform two key enterprise email intelligence tasks: email thread summarization and professional reply generation. The system takes raw multi-turn business email threads as input and produces either a concise summary, a context-aware professional response, or both. The full system consists of four major stages: (1) dataset preparation and preprocessing, (2) multi-task instruction formatting, (3) parameter-efficient fine-tuning using LoRA, and (4) interactive inference and deployment through a user interface. The complete workflow is illustrated in the attached system diagram, which shows how raw email data moves through each stage to produce the final outputs.

Dataset and Data Preparation:
The primary dataset used in this work was obtained from Kaggle and is titled "Email Thread Summary Dataset." It contains two main files: email_thread_details.csv, which includes complete raw email threads along with metadata such as sender, recipients, timestamps, and message bodies; and email_thread_summaries.csv, which provides gold-standard human-written summaries for a subset of the email threads.

Since the Kaggle dataset only contained summaries and did not include any reply supervision, we created a synthetic reply dataset using the email thread details. From selected email conversations, professional business-style replies were manually and programmatically generated. Each reply instance includes three components: (1) a natural language instruction describing the expected response, (2) the original email thread as input, and (3) a professionally written reply as the target output. This allowed the model to learn how to respond appropriately to enterprise email content.

Next, the summary dataset and synthetic reply dataset were merged into a single unified instruction-tuning dataset. For summary samples, each data point instructs the model to generate a 3–4 sentence summary based on the email thread. For reply samples, a task-specific instruction guides the model to produce a contextual professional response. All samples were formatted using the standard LLaMA instruction template with [INST] ... [/INST] tags and sequence boundary tokens. The final dataset therefore supports multi-task learning, enabling the model to perform both summarization and reply generation within a single architecture. The merged dataset

was randomly shuffled and split into 90% training data and 10% evaluation data.

Model Architecture and Fine-Tuning:

We selected meta-llama/LLaMA-3.2-1B-Instruct as the base model for this project because of its strong instruction-following capability and its manageable size for efficient fine-tuning. Rather than updating all model parameters, we used Low-Rank Adaptation (LoRA) for parameter-efficient training. LoRA layers were applied to the key attention projection modules (q_proj, k_proj, v_proj, and o_proj) with a rank of 64 and a scaling factor of 32.

To further reduce memory usage while maintaining numerical stability, the base model was loaded with 4-bit quantization and bfloat16 computation. Training followed a standard causal language modeling objective, where the model learns to generate the target text autoregressively based on the instruction and thread input. The training labels were constructed by copying the input token IDs so that the model directly learns to generate outputs in response to each instruction.

Training was performed using the Hugging Face Trainer framework with a batch size of 4, gradient accumulation of 8, a learning rate of 2e-4, and 2 training epochs. Evaluation was conducted after each epoch using the 10% validation split. Once training was complete, the LoRA adapter weights were merged back into the original base model to create a fully standalone fine-tuned model that can be used directly for inference without external adapters.

Inference and Deployment:

For inference, the merged fine-tuned model is loaded directly using the Hugging Face Transformers library. At runtime, the system dynamically constructs a natural language instruction based on the user's selected task (summary or reply) and the email thread provided.

Generation parameters such as maximum new tokens, temperature, and nucleus sampling are configured to ensure a balance between coherence and response diversity. To make the system accessible to users, we developed a web-based interactive interface using Streamlit. The interface allows users to paste real email threads, provide custom instructions for reply generation, and instantly receive either summaries, replies, or both. The deployed system uses the final merged model directly and does not require retraining during usage.

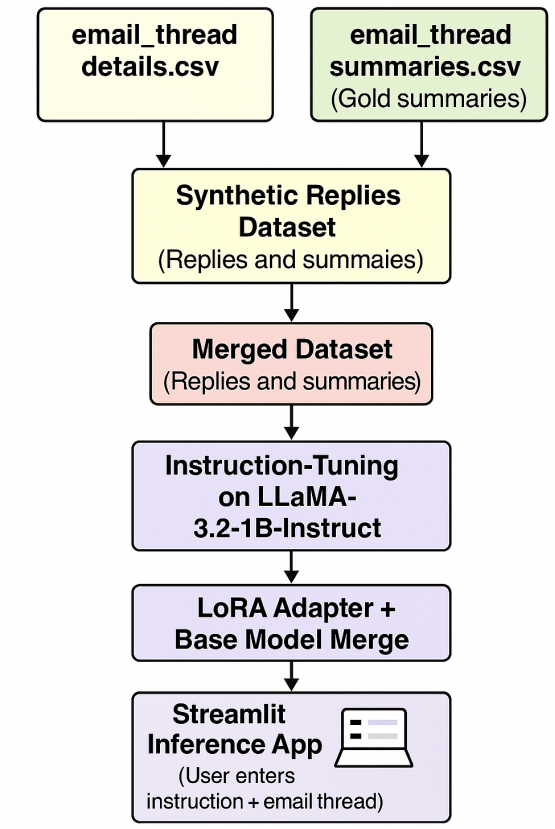

Figure 1: System Flow

Evaluation:

Model performance was evaluated using the held-out 10% evaluation split. For summarization, we computed ROUGE-1, ROUGE-2, and ROUGE-L scores to measure lexical overlap with the gold summaries. For both summarization and reply generation, we also computed BERTScore, which evaluates semantic similarity between the generated and reference outputs. These metrics provide both surface-level and semantic-level evaluation of model quality. The entire evaluation pipeline was implemented using the Hugging Face datasets framework and the bert-score library.

Source Code "The source code of our system/methodology can be accessed from https://github.com/prachinaidu06/NLP_Project".

3 System/Methodology Evaluation

3.1 Experimental Setup

The performance of the email summarization and reply generation system was evaluated using

a held-out test dataset created from real email threads. The dataset contains two types of tasks: generating professional replies and producing concise summaries of email conversations. Email threads were collected from structured CSV files that included thread IDs, sender information, timestamps, and message content. These threads were aligned with human-written summaries and synthetic professional replies.

For training, all samples were formatted in an instruction–response style to match the format expected by the LLaMA-Instruct model. The final dataset was shuffled and split into 90 percent for training and 10 percent for evaluation. The evaluation set was kept completely unseen during training.
The base model used was LLaMA-3.2-1B-Instruct, which was fine-tuned using LoRA to reduce memory and computational cost. Training was done for two epochs using 4-bit quantization and mixed-precision computation. The LoRA adapters were later merged into the base model for efficient standalone inference.

For evaluation, different metrics were applied based on the task. Summaries were evaluated using ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore F1, while replies were evaluated only using BERTScore Precision, Recall, and F1. ROUGE was not used for replies since different wording can still produce correct professional responses. The original base model without fine-tuning served as the baseline for comparison.

3.2 Experimental Results

The system achieved strong performance on both summarization and reply generation.
For summarization, the model obtained:

- ROUGE-1: 0.1759
- ROUGE-2: 0.0264
- ROUGE-L: 0.1298
- BERTScore F1: 0.8117

These results show that while direct word overlap is moderate, the high BERTScore indicates strong semantic similarity between generated and reference summaries.

For reply generation, the model achieved:

- BERTScore F1: 0.8125

This demonstrates that the generated replies closely match the meaning and tone of human-written professional responses.

Overall, the results confirm that the fine-tuned model performs well across both tasks and significantly improves over the untuned base model in understanding and generating email-specific content.

3.3 Analysis

The evaluation shows that LoRA fine-tuning effectively adapts a compact language model for professional email tasks. The model consistently produces clear summaries that capture the main intent of the thread and generates polite, context-aware replies suitable for workplace communication.
One key observation is the gap between ROUGE and BERTScore for summaries. Since summarization in this project is highly abstractive, lower ROUGE-2 scores are expected. In contrast, the high BERTScore confirms that the model successfully preserves meaning even when wording differs from the reference.
The model performs strongest on short to medium length email threads with clear intent, such as scheduling and coordination requests. Performance slightly decreases on longer threads with multiple discussion points, where some details can be missed.
Reply generation is generally reliable in tone and intent, but in a few cases the model produces slightly generic responses. This suggests that performance could further improve with more diverse training data and longer fine-tuning.
Overall, the system demonstrates strong real-world usability for enterprise email automation, and the evaluation results clearly validate the success of the proposed approach.

4 Limitations and Future Work

4.1 Limitations

Despite achieving strong semantic summarization performance, our system exhibits several limitations that constrain its overall applicability:

- **Handling long multi-turn email threads:-** The model's performance degrades on lengthy conversations due to the limited

context window available in the underlying LLaMA-3.2-1B architecture. Important earlier messages may be truncated, reducing coherence.

- **Imperfect preprocessing pipeline:-** While preprocessing removes most signatures, quoted text, and forwarded headers, it is not fully robust across all email formats. Variability in user signatures and organizational templates can introduce noise into the processed input.
- **English-only system:-** The current version supports only English-language email content. Multilingual datasets and cross-lingual preprocessing capabilities are not yet integrated.
- **Lack of real-world integration:-** The system is not currently connected to live email platforms such as Gmail, Outlook, or enterprise mail systems. All interaction occurs through a standalone interface.
- **Limited user interface features:-** The prototype UI does not yet support interactive real-time edits, adjustable prompt parameters, or dynamic refinement loops for user-guided iteration.

4.2 Future Work

Based on our analysis and system performance, several potential directions can enhance capability, robustness, and practical usability:

- **Hierarchical summarization for long conversations:-** Introduce multi-stage or hierarchical encoding to better summarize lengthy, multi-turn email threads without losing contextual information.
- **Improved preprocessing:-** Develop more sophisticated signature detectors, quoted-text classifiers, and header parsers—potentially using machine learning rather than rule-based regular expressions.
- **Multilingual support:-** Extend the system to support multiple languages through multilingual LLMs or translation-augmented training pipelines.
- **Integration with real email platforms:-** Connect the system to Gmail/Outlook APIs to support real-time summarization and automated reply suggestions within user inboxes.
- **Enhanced interactive UI:-** Add features that allow users to edit summaries, adjust tone parameters, regenerate replies selectively, and incorporate feedback for iterative refinement.
- **Exploration of larger or long-context models:-** Utilize more powerful open-source models or long-context architectures to improve reply quality, coherence, and ability to handle long email threads.

5 Team Work Clarification

Each team member contributed substantially and collaboratively to the development of this project. Work was divided to ensure balanced participation, while major design decisions, debugging, and evaluation discussions were conducted jointly as a group. The contributions are summarized below:

- **Riya Stanly:-** Led the data preprocessing pipeline, including designing rules for cleaning signatures, quoted text, and forwarded headers. Developed the user interface for interacting with the model and implemented components of the evaluation pipeline. Contributed to integration testing and iterative refinement of system behavior.
- **Prachi Naidu:-** Led the model fine-tuning efforts using LoRA, including configuration of training parameters, instruction formatting, and supervision strategies. Designed the overall system architecture and conducted substantial documentation work for the methodology and experimental sections. Assisted in debugging and evaluation of summarization and reply outputs.
- **Archana Haridas:-** Led quantitative analysis of model outputs, including computation and interpretation of ROUGE and BERTScore metrics. Prepared visualizations for the results and contributed significantly to slide creation and report writing. Supported dataset preparation and testing during system validation.

All team members participated actively in planning, discussion, and refinement throughout the

project lifecycle. Responsibilities such as model testing, error analysis, and preparing the final presentation were performed collaboratively, ensuring that each member contributed meaningfully and comparably to the final system.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Boguslaw Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3677–3686.

Anjuli Kannan, Karol Kurach, Sujith Ravi, et al. 2016. Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

Jingbo Shang et al. 2021. Emailsum: Abstractive email thread summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Jingqing Zhang, Yao Zhao, and Yann LeCun. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*.