

Title: Personalized Email Generation Assistant using LoRA-Fine-Tuned LLaMA-3 Model

Author: ABHIJEET KUMAR

Enrollment: 23324001

Institution: IIT ROORKEE

1. Introduction

This project presents a personalized **AI Email Assistant** capable of generating formal academic or professional emails automatically.

The system uses a **fine-tuned LLaMA-3.2-3B-Instruct model** to automate repetitive tasks such as writing internship or research outreach emails.

The agent combines reasoning, planning, and execution in a modular architecture — taking structured inputs (professor's name, subject, bio, and goal) and producing well-phrased formal emails with consistent tone and structure.

2. System Architecture

2.1 Overview

The architecture consists of four integrated modules:

1. **Prompt Builder** (generate_email.py)
Converts structured inputs into natural language instruction prompts.
2. **Fine-Tuned LLaMA Model** (llm_adapter.py)
Loads the base LLaMA model and applies LoRA adapters for efficient specialization.
3. **Evaluation Pipeline** (evaluate_generation.py)
Uses multiple metrics (BLEU, ROUGE-L, BERTScore, Semantic Similarity) to assess quality.
4. **Interactive Interface** (email_assistant2.py)
Provides a Command-Line Interface (CLI) for generating and saving emails interactively.

2.2 Workflow

1. User inputs → Professor details, subject, bio, and tone
2. Prompt construction → Creates structured instruction format
3. Model inference → LoRA-tuned LLaMA generates the final email
4. Post-processing → Extracts, cleans, and saves the email
5. Evaluation → Compares output with reference emails for quality assessment

This modular design allows easy retraining, evaluation, and integration with larger AI systems.

3. Fine-Tuning Setup

3.1 Model Selection Rationale

The **LLaMA-3.2-3B-Instruct** model was chosen for the following reasons:

- **Instruction-Following Ability:** Pre-trained for natural task compliance and coherent multi-sentence generation.
- **Model Size Efficiency:** The 3B parameter variant balances performance and memory usage — ideal for local LoRA training.
- **Open-Weight Availability:** Freely usable for academic research with strong multilingual reasoning capability.
- **Context Understanding:** Handles structured prompts well, making it suitable for formal email writing.

3.2 Why LoRA Fine-Tuning

LoRA (**Low-Rank Adaptation**) was selected because it enables **parameter-efficient fine-tuning**:

- Only a small fraction of model parameters are trained (low-rank updates), significantly reducing GPU memory usage.
- LoRA allows training on consumer-grade hardware without losing base model knowledge.
- It's easy to **attach and detach adapters**, making the system modular and lightweight.

This approach achieves near-full fine-tuning performance with only a few million additional parameters.

3.3 Quantization Setup

The model was fine-tuned under **4-bit quantization** to optimize training speed and memory efficiency using BitsAndBytesConfig:

Parameter	Value
load_in_4bit	True
quant_type	nf4
compute_dtype	float16
double_quant	True

This setup enabled fine-tuning a 3B model efficiently on limited hardware.

3.4 LoRA Configuration

Parameter	Value
Rank (r)	8
Alpha	16
Dropout	0.05
Target Modules	q_proj, v_proj
Optimizer	paged_adamw_32bit
Task Type	Causal Language Modeling

The model was prepared for LoRA fine-tuning via the PEFT library and trained using Hugging Face’s Trainer API.

3.5 Dataset

A **custom JSONL dataset** (train.jsonl) was used, containing *input-output pairs* of realistic academic email prompts and target responses. Each example mapped structured user input to a manually crafted formal email, teaching the model the nuances of tone, phrasing, and structure.

This dataset reflects the project’s real application domain — improving contextual accuracy in academic writing.

3.6 Training Configuration

Parameter	Value
Epochs	1

Metric	Description
ROUGE-L	Measures fluency via longest common subsequence.
BERTScore (F1)	Deep contextual alignment using transformer embeddings.

Each generated email was compared against a human-written reference, and detailed metrics were saved in timestamped JSONL logs.

5. Results and Analysis

5.1 Quantitative Results

Metric	Average Score
Semantic Similarity	0.796
BLEU	0.1681
ROUGE-L	0.3538
BERTScore (F1)**	0.9071

Samples Evaluated **5**

Interpretation:

- High **BERTScore (0.90)** and **semantic similarity (0.80)** confirm the model’s strong contextual comprehension.
- BLEU and ROUGE values are modest — expected for open-ended natural text generation tasks.
- Generated emails closely mirror the reference structure and tone while maintaining originality.

5.2 Qualitative Observations

- Model consistently produced **formal, polite, and grammatically accurate** emails.
- Maintained coherent structure: greeting → introduction → request → closing.
- Improved stylistic fluency compared to base model (less redundancy, smoother phrasing).
- Minor verbosity in some responses — can be mitigated via prompt tuning.

6. Discussion

6.1 Strengths

- High contextual accuracy achieved with **minimal compute and data**.
- LoRA tuning preserved LLaMA's general knowledge while specializing tone and phrasing.
- End-to-end pipeline (training → evaluation → generation) is modular and reproducible.

6.2 Limitations

- Dataset size limited generalization; larger corpus would enhance robustness.
- BLEU/ROUGE values show room for syntactic refinement.
- Current CLI interface lacks GUI or deployment integration.

6.3 Future Work

- Expand dataset with real academic email examples.
- Integrate into **Model Context Protocol (MCP)** for agentic reasoning.
- Add **web interface or chat UI** for broader usability.

7. Conclusion

This project successfully demonstrates the creation of a **domain-specialized AI agent** through LoRA fine-tuning of a quantized LLaMA-3 model.

The system effectively automates academic email writing — achieving high semantic accuracy, fluent generation, and stylistic control.

The results validate that **parameter-efficient fine-tuning** is a practical and resource-friendly method for adapting large language models to real-world tasks.

8. References

1. Meta AI – *LLaMA-3 Model Documentation*
2. Hugging Face Transformers – *Trainer and PEFT Framework*
3. Dettmers et al. – *BitsAndBytes Quantization for LLMs*
4. Hu et al. – *LoRA: Low-Rank Adaptation of LLMs*
5. SentenceTransformers – *Semantic Text Similarity Models*
6. Zhang et al. – *BERTScore: Evaluating Text Generation with BERT*
7. Lin – *ROUGE: Recall-Oriented Evaluation Metrics for Summarization*