Email Summarization & Reply Agent

Team Lead: MALLIKESWARAPU THANUJA [22B21A4301]

Team Members:

BANDLAMUDI ANKALAIAH	[22B21A4352]
GANUGULA SIVA SANKAR	[22B21A4334]
ARIGE SHIVA RAMAKRISHNA	[22B21A4356]
BHIMANA DEVID RAJU	[22B21A43A5]

Department: CSE (Artificial Intelligence)
Institution: KIET
Academic Year: 2025–26

Abstract

This project develops an intelligent system that summarizes long emails and generates smart replies.
With the rapid growth of digital communication, people often spend significant time reading lengthy emails and formulating responses. This project addresses the issue by employing Transformer-based models such as **BART-large** for summarization and **T5/GPT** models for reply generation.

The system reduces email reading time, improves productivity, and provides smart communication assistance. The backend is implemented using **Python and Flask**, while **Hugging Face Transformers** provide the core AI models. The frontend uses HTML, CSS, and JavaScript to interact with users.

The expected outcome is a scalable and efficient **AI-powered email assistant** that can summarize content concisely and suggest meaningful replies.

Introduction & Problem Statement

Background

Emails remain one of the most widely used means of communication in academic, professional, and personal contexts. However, the increasing number of emails results in **email overload**, which wastes time and reduces efficiency.
Reading each message in detail and preparing responses is often repetitive and time-consuming.

Importance of the Problem

- Employees and students spend a considerable portion of their day on emails.
- Delayed or missed communication can lead to productivity losses.
- Automation can support faster responses and decision-making.

Problem Definition

The project seeks to solve two core challenges:

1. **Summarization**: Automatically condense lengthy emails into concise, informative summaries.
2. **Reply Generation**: Suggest appropriate and smart replies, reducing manual effort.

Objective

To design a **lightweight AI-powered tool** that integrates summarization and reply generation into a seamless system for efficient communication.

Proposed Methodology

Approach

The project leverages **Natural Language Processing (NLP)** with Transformer models. These models excel at understanding text and generating human-like language. The approach involves:

- **Summarization** using **BART-large**, pretrained on the CNN/DailyMail dataset.
- **Reply Generation** using **T5 or GPT-based models**, trained for conversational tasks.

Own model (scratch)

---

1. Collect & Clean Email Data

- Gather real email datasets (e.g., Enron).
- Extract raw emails, summaries, and replies.
- Clean the text: remove HTML, signatures, and quoted replies.

---

2. Build a Custom Tokenizer

- Create your own tokenizer (word-level or character-level).
- Build a vocabulary from scratch.
- Convert text to token IDs and handle unknown/padding tokens.

---

3. Design Your Model Architecture

- **Summarization**: Build a **Transformer Encoder-Decoder**.

- **Reply Generation**: Build a **Decoder-only Transformer**.
- Implement attention, positional encoding, and feedforward layers.

---

4. Prepare Training Pipeline

- Create training samples with input/output token sequences.
- Add masks for padding and future tokens.
- Use cross-entropy loss and the Adam optimizer.

---

5. Train Your Model

- Train on your custom dataset using your custom model.
- Monitor training loss and sample outputs.
- Save model checkpoints after epochs.

---

6. Build Your Own Evaluation Metrics

- Implement **ROUGE** (for summaries) and **BLEU** (for replies) from scratch.
- Compare generated vs. reference outputs using token overlaps.
- Evaluate regularly to check progress.

---

7. Inference & Deployment

- Write your own decoding logic (greedy or beam search).
- Generate summaries and replies from new email inputs.
- Integrate into a script, CLI tool, or basic UI for usage

Tools & Technologies

- **Python** – core programming language
- **Flask** – backend framework to handle requests and responses
- **Hugging Face Transformers** – for state-of-the-art NLP models
- **HTML, CSS, JavaScript** – frontend interface for user interaction

Dataset

- Pretrained summarization models like **BART** trained on **CNN/DailyMail** dataset.
- Reply generation using **T5/GPT** adapted to conversational email data.
- Transfer learning ensures minimal additional training effort.

Flowchart (Vertical Representation)

User

↓

Input Email

↓

Flask Backend

↓

Summarization (BART-large)

↓

Reply Generator (T5/GPT)

↓

Output: Summary + Suggested Replies

Working of the System

1. The **user enters or uploads** an email text on the frontend.
2. The **Flask backend** receives the input and forwards it to the summarizer model.
3. The **summarizer** generates a short and meaningful summary.
4. The **reply generator** proposes multiple smart responses.
5. The **frontend** displays both the summary and replies clearly to the user.

This ensures reduced email reading time, enhanced productivity, and seamless communication.

Conclusion

The **Email Summarization & Reply Agent** demonstrates how Artificial Intelligence can transform communication.
By combining summarization and smart reply generation, the system:

- Reduces email overload
- Enhances decision-making
- Saves time and effort
- Provides a scalable tool for academic, professional, and enterprise use

This AI-driven approach can be extended further into chat platforms, customer service bots, and corporate communication systems.

References

---

1. Hugging Face Transformers: https://huggingface.co

2. Lewis, M. et al. (2020). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation.*
3. Raffel, C. et al. (2020). *T5: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.*
4. Vaswani, A. et al. (2017). *Attention is All You Need.*
5. Flask Documentation: https://flask.palletsprojects.com