AI Email Agent: Architecture Document

Navyatha Chunduri, IIT Hyderabad

September 17, 2025

---

1 Overview

The AI Email Agent is a Python-based application designed to automate email processing. It connects to a user's Gmail account, fetches unread emails, and uses a two-stage AI pipeline to classify them for importance and summarize the critical ones. The goal is to reduce manual email triage and provide the user with immediate, concise summaries of what matters most.

2 Core Components

The agent's architecture consists of three primary components, detailed in the table below.

Component	Technology	Purpose
Data Ingestion	Google Gmail API	To securely authenticate and fetch unread emails from the user's inbox in real-time.
Classification Module	Fine-tuned Hugging Face Transformer	A locally-hosted text classification model that analyzes email content to predict its importance.
Summarization Module	Google Gemini API (Few-Shot Prompting)	A cloud-based Large Language Model (LLM) that generates concise summaries of important emails.

Table 1: Core Architectural Components

3 Models Used & Design Rationale

3.1 Classification Model

- **Model:** The model chosen for the classification task was bert-base-uncased.
- **Reason for Choice:**
  - **Dataset Constraints:** The training dataset was limited to **500 emails** due to constraints in the data acquisition process. This small, specialized dataset is insufficient for fine-tuning a Large Language Model (LLM) effectively, as LLMs require much larger datasets to learn and avoid overfitting.

- **BERT's Suitability for Smaller Datasets:** Models like BERT are pre-trained on vast amounts of general text and can be effectively fine-tuned on smaller, domain-specific datasets. For a dataset in this range, BERT-variants are an excellent choice as they can learn the specific patterns, keywords, and context that define an "important" email for the user.
- **Efficiency:** As a lightweight, local model, it makes classification extremely fast and cost-effective, avoiding the need for an API call for every incoming email.

3.2 Summarization Model

- **Model:** 'gemini-2.5-flash' via the Google Generative AI API.
- **Reason for Choice:**
  - **High-Quality Summaries:** State-of-the-art LLMs like Gemini excel at understanding context and generating human-like, abstractive summaries. Training a high-quality summarization model from scratch would require a massive dataset and significant resources.
  - **Flexibility with Few-Shot Prompting:** The agent uses a few-shot prompting technique. This allows it to guide the powerful Gemini model to produce summaries in a specific, desired format without any additional training. This is a highly efficient way to achieve customized, high-quality output.

Interaction Flow & Data Pipeline

The agent operates on a clear, sequential data pipeline for each processing cycle.

- **Authentication:** The script first calls `get_gmail_service()` to authenticate with Google using OAuth 2.0. The first time, it prompts user login via a browser; on subsequent runs, it uses the saved `token.json` for seamless access.
- **Fetch Unread Emails:** The `fetch_unread_emails()` function is called, which connects to the Gmail API and retrieves a list of all unread messages from the inbox.
- **Iterate and Classify:** The agent loops through each fetched email. For every email, its body is passed to the `classify_email_as_important()` function.
- **Conditional Logic Gate:** The boolean result from the classifier acts as a gate:
  - If **False**, the email is deemed unimportant. The agent prints a "No action needed" message and moves to the next email.
  - If **True**, the email is deemed important and is passed to the next stage.
- **Summarize:** The `get_email_summary()` function is called with the important email's content. This function constructs a detailed few-shot prompt and sends it to the Gemini API.
- **Output Action:** The summary received from the API is printed to the console under an "ACTION REQUIRED" heading, presenting the final, actionable output to the user.

The agent operates as illustrated in the flowchart below.

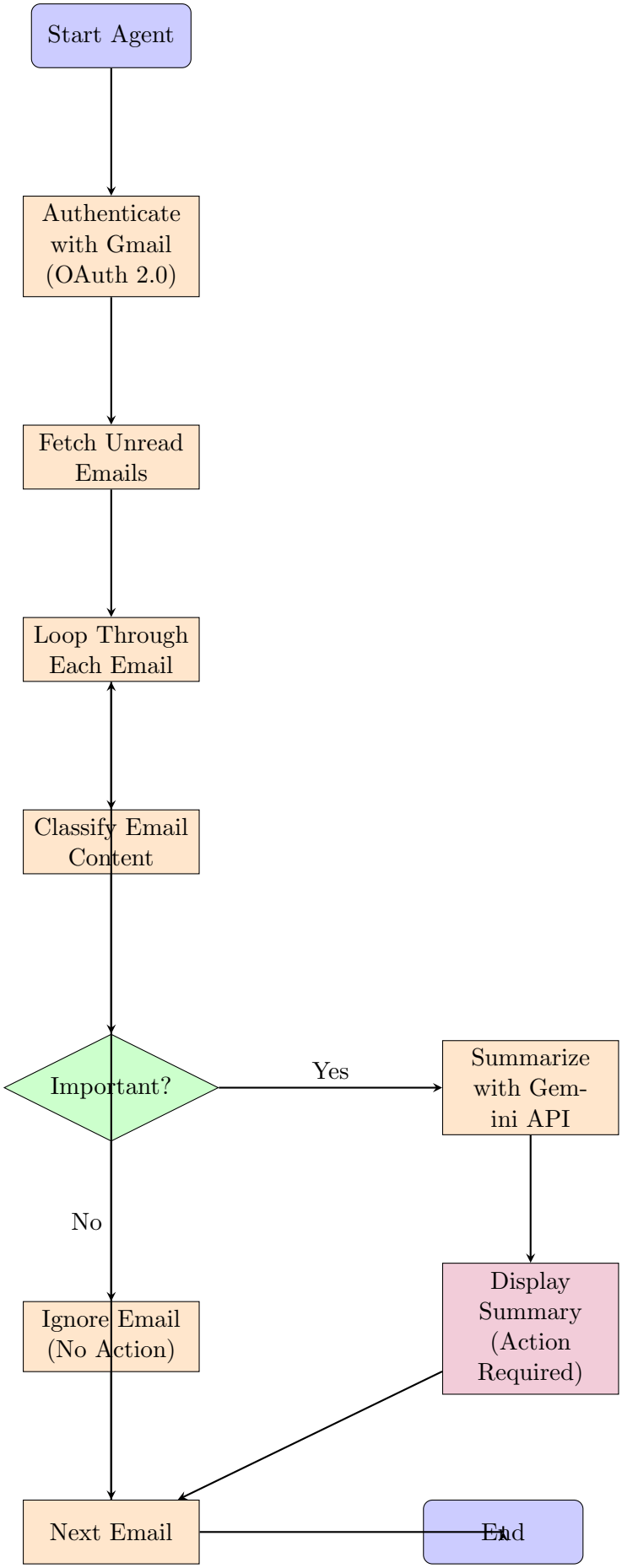

Figure 1: AI Email Agent Workflow