Data Science Report: Fine-Tuning and Evaluation of Email Automation Models

AI Email Agent Project

Ashwin Gaikwad - IIT Goa

September 16, 2025

Abstract

This report documents the data science processes behind the two core machine learning models in the AI Email Agent system. It details the systematic data pipeline, fine-tuning setup, and the robust evaluation methodologies used to validate the performance of Model 1 (a DistilBERT-based email classifier) and Model 2 (a Llama 3-based data extractor).

1 Model 1: DistilBERT Email Classifier

This model serves as the initial triage agent, responsible for rapidly categorizing all incoming emails.

1.1 Fine-tuning Setup

1.1.1 Data Preparation Pipeline

A multi-step process was used to transform raw emails into a clean, labeled dataset ready for training.

- **1. Data Ingestion**: Over 5,600 emails were fetched from a personal Gmail account using the `fetch_emails.py` script.
- **2. Data Cleaning**: Each email was rigorously cleaned using `clean_email_dataset.py`, which removed HTML tags, URLs, and normalized whitespace to reduce noise.
- **3. AI-Powered Labeling**: The cleaned emails were labeled into 8 categories using the Gemini API via the `classify_dataset_with_gemini.py` script, which employed a detailed, rule-based prompt.
- **4. Final Preparation**: The dataset was shuffled (`shuffle_data.py`) and prepared (`prepare_data.py`) by combining sender, subject, and body fields into a single input.

1.1.2 Training Configuration

- **Base Model**: `distilbert-base-uncased`.
- **Dataset Split**: The prepared data was split into an 80% training and 20% test set.
- **Training Hyperparameters**: The model was fine-tuned for 3 epochs with a batch size of 8.

1.1.3 Results Summary

The fine-tuned model achieved excellent results on the validation set, with an **Overall Accuracy of 96.1%** and a **weighted F1-Score of 0.96**.

1.2 Evaluation Methodology and Outcomes

A comprehensive evaluation was performed by the `evaluate_distilbert_classifier.py` script on the held-out 20% test set. The script iterates through each test example, generates a prediction, and compares it against the ground-truth label.

- **Classification Report**: Beyond a single accuracy score, a full classification report was generated. This provides a granular view of performance, detailing the precision, recall, and F1-score for each of the 8 individual categories, which is crucial for understanding class-specific strengths and weaknesses.

- **Confusion Matrix**: To visually diagnose misclassifications, a confusion matrix was generated and saved as a heatmap image. This allows for easy identification of which categories the model tends to confuse with one another.

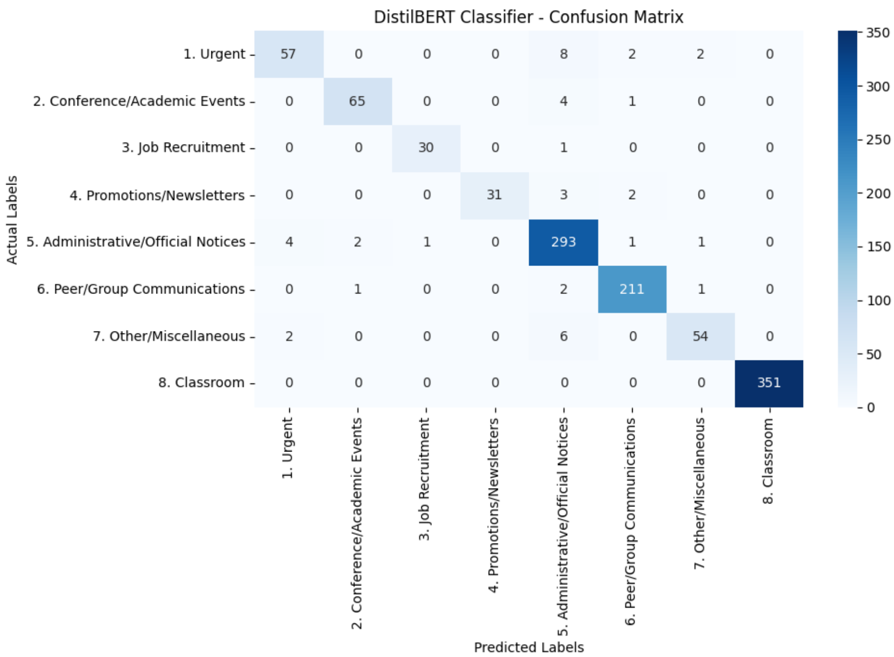

Figure 1: Confusion matrix heatmap generated by the evaluation script, showing the classifier's performance across all 8 categories.

2 Model 2: Llama 3 Data Extractor

This model acts as a specialist agent, responsible for parsing text from emails and extracting structured data in a JSON format.

2.1 Fine-tuning Setup

2.1.1 Training Configuration

- **Base Model**: `meta-llama/Meta-Llama-3-8B-Instruct`.

- **Training Data**: A custom `.jsonl` file where each entry consisted of an email's text paired with a hand-crafted, high-quality JSON output.
- **Fine-tuning Method**: Low-Rank Adaptation (LoRA) was used for memory-efficient training, along with 4-bit quantization.

2.1.2 Results Summary

The evaluation showed perfect structural adherence but room for improvement in factual accuracy. The **JSON Validity Rate was 100%**, the **Exact Match Rate was 26.3%**, and the more representative **Field-Level F1-Score was 0.69**.

2.2 Evaluation Methodology and Outcomes

A custom evaluation approach was implemented in the `evaluate_extractor.py` script to assess the quality of the generative output on a 10% test split.

- **Evaluation Process**: The script runs inference on each test email, uses a safe parsing function to extract the generated JSON, and compares it against a ground-truth JSON.
- **Performance Metrics**: The evaluation used three key metrics to provide a comprehensive view of performance:
  - **JSON Validity Rate**: Measures if the model produces syntactically correct JSON. This is a pass/fail test for basic structural integrity.
  - **Exact Match Rate**: A strict metric that checks if the entire generated JSON is a perfect, character-for-character match with the ground truth.
  - **Field-Level F1-Score**: A more nuanced metric that calculates the precision and recall on individual key-value pairs. This indicates how well the model extracts the correct information, even with minor formatting differences.

3 Key Findings

- **Systematic Pipeline is Key**: The comprehensive and sequential data pipeline—from fetching and intensive cleaning to sophisticated AI-powered labeling—was fundamental to the high performance of the classifier.
- **Tailored Evaluation is Crucial**: The project successfully employed two different evaluation strategies tailored to the model types: detailed classification metrics and a confusion matrix for DistilBERT, and custom JSON-parsing metrics for the generative Llama 3 model.
- **Opportunity for Extractor Improvement**: While the Llama 3 extractor reliably produces valid JSON, its Field-Level F1-Score of 0.69 indicates room for improvement. Future iterations would benefit from a larger, more meticulously hand-crafted training dataset, more extensive fine-tuning, or potentially using a more complex base model.