The scripts require the following Python libraries. You can install them using pip:

pip install Flask python-dotenv py-email scikit-learn PyPDF2 python-docx google-generativeai

Specifically, the scripts use:

* **Flask:** For creating the API endpoints.
* **python-dotenv:** For loading environment variables from a .env file.
* **py-email:** (email module) For parsing email messages.
* **PyPDF2:** For extracting text from PDF files.
* **python-docx:** (docx2txt) For extracting text from DOCX files.
* **scikit-learn:** (sklearn) For text vectorization (TfidfVectorizer) and calculating cosine similarity for duplicate detection.
* **google-generativeai:** For interacting with the Google Gemini API for email classification, data extraction, and intent detection.
* **logging:** For logging errors and warnings.
* **os:** For interacting with the operating system, such as file path manipulation.
* **json:** For working with JSON data, especially in the API endpoints.
* **numpy:** (Implicitly used by scikit-learn) For numerical operations.
* **langchain-google-genai:** (Used in one of the sources as an alternative way to interact with Gemini).

**Installation**

1. **Clone the repository**
2. **Install the dependencies** listed above using pip:
3. pip install -r requirements.txt
4. **Set up environment variables:**
   * Create a .env file in the project root.
   * Add your Google Gemini API key to the .env file:
   * GOOGLE\_API\_KEY=YOUR\_GOOGLE\_API\_KEY

**Usage**

The toolkit can be used in two main ways: as a set of Python functions or through the provided Flask API.

**Using Python Functions Directly**

You can import and use the individual functions in your own Python scripts. For example:

from main import process\_email, detect\_duplicate\_emails

email\_path = "path/to/your/email.eml"

request\_types = ["Adjustment", "Fee Payment", "AU Transfer", "Money Movement-Outbound", "Commitment Change", "Money Movement-Inbound", "Closing Notice"]

**Using the Flask API**

1. **Run the Flask application:** Navigate to the directory containing the app.py:
2. **Process a single email:** Send a POST request to the /process\_email endpoint with the email file attached as email\_file. For example, using curl:

*curl -X POST -F "email\_file=@path/to/your/email.eml" http://127.0.0.1:5000/process\_email*

The API will return a JSON response containing the extracted information.

1. **Detect duplicate emails:** Send a POST request to the /detect\_duplicates endpoint with a JSON payload containing a list of email texts under the key email\_texts. For example, using curl:
2. curl -X POST -H "Content-Type: application/json" -d '{"email\_texts": ["text of email 1", "text of email 2"]}' http://127.0.0.1:5000/detect\_duplicates

**Configuration**

* **Attachment Output Directory:** The directory where email attachments are saved is currently set to "C:/Users/TaskTurners/Downloads/TaskTurners/data/Attachment/" in some scripts and "attachments" in another. You may want to configure this based on your needs.