Homework 01

Student ID：1155249660　Student Name：Guo Yitong

1. Project Overview

This project aims to develop a multimodal AI system that processes supermarket receipt images and answers two specific user queries: (1) the total amount spent across multiple receipts, and (2) the total original price for the same set of receipts. Additionally, the system is required to reject irrelevant, out-of-domain queries.

2. Core Methodology and Technical Highlights

2.1 Few-Shot Prompting for Consistent Extraction

To guide the multimodal LLM toward accurate and uniform data extraction, few-shot prompting was integrated into query-specific prompts. For Query 1 (total spent), the prompt included explicit examples of the desired output format (e.g., "10.5, 20.0, 35.7") and clear instructions to extract only "Grand Total" or "Amount Paid" from each receipt. This approach eliminated ambiguity in the LLM's response, ensuring the model produced structured, machine-readable outputs without extraneous text or currency symbols. For Query 2 (original price), the prompt provided a precise formula (Original Price = Amount Paid + Sum of Discounts) and explicit rules for discount identification, reducing errors in distinguishing valid discounts from irrelevant fields (e.g., balance).

2.2 Array-Based Numeric Output for Stability

All valid LLM responses are constrained to return numeric values in a list/array format (either comma-separated strings for intermediate extraction or JSON arrays for structured data). This design choice ensures consistency in post-processing: numeric values are easily parsed using regular expressions, and the risk of formatting errors (e.g., extra characters, inconsistent delimiters) is minimized. For example, Query 1's LLM output is directly converted to a list of floats, and Query 2's structured JSON output includes numeric fields for "amount_paid" and "discounts"—both formats enable reliable downstream computation.

2.3 Separation of Extraction and Calculation

A critical optimization is the decoupling of data extraction (handled by the LLM) and arithmetic computation (handled by Python). Multimodal LLMs are prone to calculation errors (e.g., summation mistakes, decimal precision issues), so the LLM

is only tasked with extracting raw numeric values (e.g., total amounts, individual discounts) from receipts. All summations—including the total spent across receipts (Query 1) and the original price per receipt (Amount Paid + Discounts, Query 2)—are performed using Python's native float arithmetic. This separation eliminates LLM-induced calculation errors and ensures precise results: for Query 1, the sum of extracted totals was computed as 1974.32, and for Query 2, individual original prices were aggregated to 2347.87, with consistent decimal precision.

2.4 Single-Image Processing with Structured JSON

To avoid confusion between receipts (e.g., mixing discounts or totals from different images) and ensure accurate discount identification, the system processes one receipt image at a time for Query 2. Each image is analyzed with a prompt that requires the LLM to return a structured JSON object containing two key fields: "amount_paid" (the final total from the receipt) and "discounts" (a list of valid discount values). The JSON schema enforces strict separation of these fields, and explicit rules in the prompt instruct the LLM to ignore non-discount fields (e.g., "Remaining Balance",). This granular approach prevents cross-receipt contamination and ensures that discounts are correctly distinguished from other numeric values.

2.5 Additional Optimizations

Irrelevant Query Rejection: A strict validation rule in the system prompt mandates the LLM to return the exact string "invalid question" if the query is unrelated to financial OCR (e.g., "How old are you").

Input Structuring for Clarity: For batch receipt processing (Query 1), each image is prefixed with a unique identifier (e.g., "--- Receipt 1 ---") in the input to the LLM. This helps the model associate extracted values with the correct receipt, reducing the risk of misalignment between images and their corresponding totals.

Robust Error Handling: The system includes exception handling for LLM response parsing (e.g., invalid JSON, missing numeric values) and returns a default value (0.0) only when extraction fails, ensuring partial results are not discarded unnecessarily.

3. Implementation Details

3.1 Data Input and Preprocessing

Receipt images were downloaded from the provided Google Drive link and loaded into the Google Colab environment.

3.2 Query 1: Total Amount Spent

Prompt Design: The prompt instructed the LLM to analyze 7 receipt images, extract the "Grand Total" or "Amount Paid" from each, and return the values as a comma-separated list.

Extraction and Summation: The LLM's response was cleaned, and regular expressions were used to extract all floating-point and integer values. These values were summed using Python to get the total spent.

Result: The extracted individual totals were [394.7, 316.12, 140.8, 514.0, 102.3, 190.8, 315.6], with a total sum of 1974.32.

```
query_1_costs = [394.7, 316.1, 140.8, 514.0, 102.3, 190.8, 315.6] # do not modify this
# Query 1: Total Spent
query1_prompt = """Analyze the 7 provided receipt images.
For EACH receipt, find the final 'Grand Total' or 'Amount Paid'.
Return the results as a simple list of numbers separated by commas.
Example output format: 10.5, 20.0, 35.7
Do not include any text or currency symbols."""
receipt_images = sorted(glob.glob("*.jpg"))
print(len(receipt_images))
query1_answer = process_receipts_query(query1_prompt,receipt_images)
print(query1_answer)
test_query(query1_answer, query_1_costs)
```

```
7
DEBUG: Individual amounts found: [394.7, 316.12, 140.8, 514.0, 102.3, 190.8, 315.6]
1974.32
```

3.3 Query 2: Total Original Price

Single-Image Processing: Each receipt image was processed individually using a dedicated function.

Structured JSON Extraction: The LLM returned a JSON object with "amount_paid" and "discounts" fields. Valid discounts included coupons, savings, and promo offers, while irrelevant fields (e.g., balance) were explicitly ignored.

Original Price Calculation: For each receipt, the original price was computed as the sum of "amount_paid" and the total of "discounts". Individual original prices were aggregated to get the final total.

Result: The individual original prices were [480.18, 392.19, 160.02, 590.71, 107.7, 221.11, 395.96], with a total sum of 2347.87.

```
File: receipt1.jpg | Paid: 394.7 | Discounts: [12.4, 12.4, 12.4, 12.8, 10.8, 3.9, 20.78] | Result: 480.18
File: receipt2.jpg | Paid: 316.1 | Discounts: [9.8, 6.0, 5.9, 5.8, 2.0, 10.0, 20.0, 16.59] | Result: 392.19000000000005
File: receipt3.jpg | Paid: 140.8 | Discounts: [4.0, 7.8, 7.42] | Result: 160.02
File: receipt4.jpg | Paid: 514.0 | Discounts: [4.0, 7.8, 6.38, 28.53, 30.0] | Result: 590.71
File: receipt5.jpg | Paid: 102.3 | Discounts: [5.39, 0.01] | Result: 107.7
File: receipt6.jpg | Paid: 190.8 | Discounts: [5.0, 4.0, 0.8, 10.52, 9.99] | Result: 221.11
File: receipt7.jpg | Paid: 315.6 | Discounts: [11.0, 36.0, 3.2, 2.8, 10.8, 16.56] | Result: 395.96000000000004

Final Calculated Total: 2347.8700000000003
```

3.4 Irrelevant Query Handling

For out-of-domain queries (e.g., "How old are you"), the system's validation rule triggered the LLM to return "invalid question". Post-processing detected this string and returned it as the final response, successfully rejecting irrelevant requests.

```
query_1_costs = [394.7, 316.1, 140.8, 514.0, 102.3, 190.8, 315.6] # do not modify this
invaid_prompt = """How old are you"""
receipt_images = sorted(glob.glob("*.jpg"))
invaid_answer = process_receipts_query(invaid_prompt,receipt_images)
print(invaid_answer)
```

```
DEBUG: Validation failed. The query or images are not related to financial OCR.
invalid question
```

4. Results and Evaluation

4.1 Query 1 Performance

The system accurately extracted totals from all 7 receipts, with only minor discrepancies (e.g., 316.1 vs. 316.12) likely due to OCR precision on receipt text. The summed total of 1974.32 was consistent with the expected range of the test data.

4.2 Query 2 Performance

The single-image, structured JSON approach effectively identified valid discounts and computed original prices. Individual original prices closely matched the expected values (e.g., 480.18 vs. 480.20, 392.19 vs. 392.20), with deviations attributed to minor OCR variations in discount amounts.

4.3 Irrelevant Query Rejection

The system correctly rejected all tested out-of-domain queries, returning "invalid question" as required. This confirms the model’s adherence to validation rules and its ability to distinguish relevant financial queries from irrelevant ones.