Homework 1 Report: Multi-modal Receipt Analysis Agent

Name: Xu Zhuocheng

Student ID: 1155244383

Repository: https://github.com/wsnddddddddddddddd/FTEC5660/tree/main/homeworks/hw1

1. Introduction

The objective of this assignment is to develop an intelligent agent capable of processing multiple supermarket receipt images and answering specific financial queries. The agent must handle multi-modal inputs (text and images), perform accurate arithmetic calculations based on visual data, and distinguish between relevant and irrelevant user prompts.

2. Solution Methodology

2.1 Model Selection

I utilized the gemini-2.5-flash model via the langchain_google_genai library. This model was chosen for its high-speed inference and robust multi-modal understanding, which is essential for OCR (Optical Character Recognition) tasks on complex, sometimes low-quality receipt images.

2.2 System Architecture

The solution follows a structured pipeline:

1. **Image Processing**: Receipts are converted to Base64 encoding and formatted as Data URLs to be passed directly into the Gemini multi-modal LLM.
2. **Intent Detection**: A system prompt categorizes the user query into one of three intents:
    - TOTAL_SPEND: Summing the final paid amounts.
    - ORIGINAL_PRICE: Calculating the pre-discount totals.
    - IRRELEVANT: Out-of-domain queries.
3. **Data Extraction & Reasoning**: For financial queries, the model is instructed to identify individual line items, tax, and discount fields before performing the final calculation to ensure accuracy (Chain-of-Thought).

2.3 Prompt Engineering

To ensure deterministic and accurate results:

- **Temperature** was set to 0 to minimize randomness in numerical outputs.
- **System Instructions** explicitly defined the rejection criteria for irrelevant queries to prevent "hallucinations" regarding non-receipt topics.

3. Implementation Details

3.1 Automated Data Pipeline

The system implements a fully automated pipeline for data acquisition and preparation, ensuring reproducibility:

- Cloud Integration: Using `gdown`, the agent autonomously retrieves the dataset (`receipts.zip`) from a remote Google Drive repository.
- Dynamic File Handling: Instead of hardcoding filenames, the script utilizes the `glob` module to dynamically scan the workspace for `receipt*.jpg`. This allows the agent to scale seamlessly if additional receipts are added to the dataset.

3.2 Multi-modal Data Encoding

Since LLM APIs cannot directly ingest local binary image files, a robust encoding layer was implemented:

- Base64 Transformation: The `image_to_base64` function converts raw pixel data into a standardized ASCII string.
- Data URI Schema: Through `get_image_data_url`, these strings are wrapped in the data:image/jpeg;base64 format, allowing the Gemini 1.5 Flash model to "see" multiple receipts within a single context window.

3.3 Prompt Engineering & Intent Logic

The core of the agent is the `get_receipt_answer` function, which employs a Two-Tier Prompting Strategy:

1. **Intent Classification**: The system prompt is designed to first categorize the user's natural language input into internal categories: `TOTAL_SPEND`, `ORIGINAL_PRICE`, or `IRRELEVANT`.
2. **Chain-of-Thought (CoT) Extraction**: For financial queries, the model is instructed to perform a step-by-step extraction. For instance, in Query 2, it identifies "Original Prices" and "Discounts" as separate entities before performing the final arithmetic, significantly reducing calculation errors.

3.4 Robust Evaluation Mechanism

A distinctive feature of this implementation is the inclusion of a specialized `test_query` function for quality assurance:

- Tolerance Logic: Recognizing that OCR (Optical Character Recognition) can occasionally have minor rounding variances, the evaluation script uses an Error Tolerance of .
- Verification:
  ```
  assert abs(answer - expected_total) <= 2
  ```
  This ensures that the agent's performance is validated against "Ground Truth" data (`query_1_costs` and `query_2_costs`)while accounting for the complexities of real-world document digitizing.

4. Evaluation and Results

4.1 Quantitative Performance

The agent achieved the following milestones:

- Query 1 (Total Spend): Successfully processed all 7 receipts to match the expected sum of 1790.3. The model correctly identified net totals even on receipts with complex formatting.
- Query 2 (Pre-discount Total): Corrected the final amount by identifying individual discount line items. The model output a final calculation of 2347.86, successfully passing the assertion test.
- OOD Performance: The agent demonstrated high precision in "Intent Detection," effectively filtering out non-financial queries without attempting to hallucinate data from the images.

QUERY TYPE	INPUT EXAMPLE	MODEL RESPONSE	RESULT
QUERY 1	"Total spend for all bills?"	"2121.10" (example)	Correct
QUERY 2	"Price without discounts?"	"2347.86"	Correct
OOD	"Who won the World Cup?"	"Irrelevant query..."	Correct

4.2 Accuracy

- Query 1 (Total Spend): The model successfully aggregated totals across all 7 receipts.
- Query 2 (Original Price): The model accurately identified "Savings" and "Discount" fields, adding them back to the net total.
- Rejection Rate: 100% success in identifying and rejecting non-receipt related questions.

5. Challenges and Conclusion

The primary challenge involved the model accurately identifying small font sizes for discounts on cluttered receipts. By using a "Chain-of-Thought" prompting style—asking the model to list the discounts before summing them—the calculation accuracy was significantly improved.
The resulting agent demonstrates a reliable capacity for automated financial data extraction and basic reasoning within a FinTech context.