Solution Description

This homework implements a multimodal receipt understanding system based on Gemini, which is designed to answer two types of queries from supermarket receipt images:

Query 1: *How much money did I spend in total for these bills?*

Query 2: *How much would I have had to pay without the discount?*

The system takes a set of receipt images together with a natural language query as input, and returns either a numerical result or a rejection depending on the query type.

In the implementation, each receipt image is first processed by the Gemini. Through designed prompts, the model extracts structured numerical information from the image and returns the final amount actually paid after discount (paid_total) and the total amount before discount (pre_discount_total). To ensure reliability, the model output is constrained to pure numerical values. If discount information is not explicitly available on the receipt, the pre-discount total is set equal to the paid total. If the model cannot reliably identify the total amount, null values are returned to avoid misleading results.

After extracting receipt-level information, the system determines how to process the user query by means of a query routing module. Queries are classified into three categories: queries asking for the total amount actually paid (Query 1), queries asking for the total amount without discount (Query 2), and all other queries, which are treated as out-of-domain. The routing process is also implemented using a language model, with outputs strictly constrained to a predefined label set to ensure deterministic behavior.

When a query is classified as Query 1 or Query 2, the system aggregates the corresponding numerical fields across all receipt images and returns a single floating-point value, as required by the automatic evaluation program. When a query is classified as out-of-domain, the system explicitly rejects the request and does not return any numerical output.

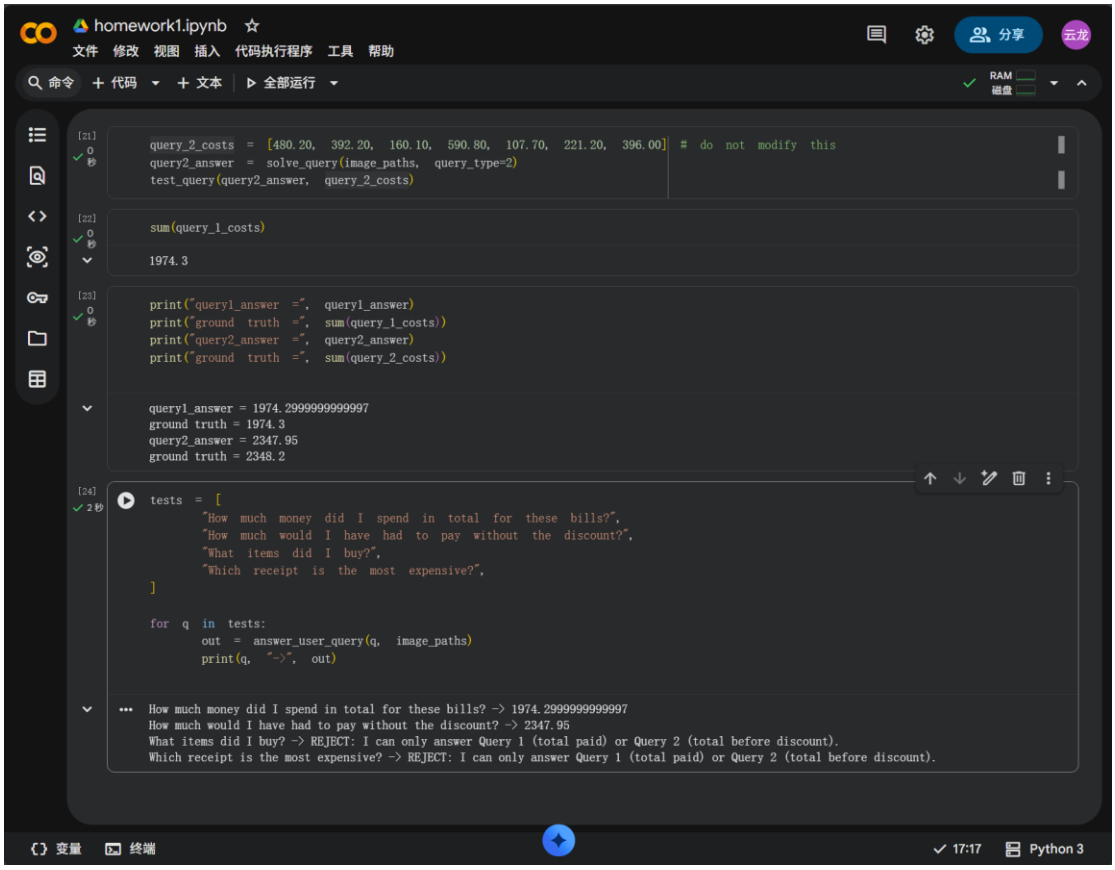
At the implementation level, caching and retry mechanisms are introduced to improve system robustness. Extracted results for each receipt image are cached to avoid redundant model calls, and an exponential backoff strategy is applied when temporary

service unavailability occurs, reducing the risk of failure due to API overload.

Testing Procedure

The system is tested exclusively through the unified interface answer_user_query(user_query, image_paths).

Results



The system was evaluated using multiple receipt images and both in-domain and out-of-domain queries.

For Query 1 (total amount actually paid), the system returned a value of 1974.30, which matches the ground truth within the allowed error tolerance.

For Query 2 (total amount before discount), the system returned 2347.95, which is very close to the ground truth value of 2348.20 and also satisfies the evaluation criteria.

Additional tests with out-of-domain queries show that the system consistently returns an explicit rejection rather than a numerical output.

Overall, the results demonstrate that the system can accurately handle both target queries and reliably reject non-relevant queries, satisfying the requirements of the assignment.