Invoice/Bill Extraction using Gemma (Multimodal LLM Agent)

Testing Summary

- **Images tested**: 5 real invoice images
- **Questions asked**: 50+ natural language queries
- **Observed Accuracy**: ~around 95-100% (consistently accurate answers for all test cases)

Folder structure

```
project_root/
│
├── images/
│   └── [Tested invoice images]
│
├── result/
│   ├── image_json.json        # Extracted structured JSON from the invoice image                    (image - name of image in images folder
│   └── image_history.json     # Q&A history with the agent
│
├── Code/
│   └── BILL_analytics.ipynb   # Main notebook containing the entire pipeline
```

Technical overview

Model - gemma (LLM)

Reason - Selected for its multimodal capabilities, enabling accurate extraction of information from images such as invoices or bills.

Prompt technique - ReAct prompting

Reason -

- Enables the agent to iteratively reason and decide when to use tools.

- Useful for complex invoice-based queries where structured extraction and logical interpretation are required.
- Helps constrain responses to invoice-specific content.

Tools

1. `Calculator`: For evaluating expressions like “500 * 0.9” or “20% of 300”.
2. `tavily_search`: A search engine tool to fetch external information if required.
3. `StringLengthCalculator`: Calculates the character length of any string (e.g., invoice IDs or line items).
4. `GetRawJSON`: Returns raw JSON data extracted from the image.

5. `JSONQueryTool`: Enables querying structured JSON with natural language.

Code overview

- **Setup**
  - Install required libraries.
  - Install and run `ollama` with `ollama serve`.
  - Pull the `gemma-12b` model.

- **LMM (Large Multimodal Model) Initialization**
  - If the input image is large, it’s split horizontally with a 100-pixel overlap, processed individually, and then results are combined.
  - For standard-size images, Gemma directly extracts invoice details.

- **Agent Creation**
  - Constructs an agent using:
    - The Gemma model
    - ReAct-style prompting
    - The defined tools
    - A result parser

- **Agent Execution**

  - Accepts a user image and a query.

- Iterates up to 10 steps, using tools as needed to produce an accurate answer.
- Final response is stored in a structured format.

How to run the code

Go to last 4 cells

1. Provide the path to the input image.
2. Gemma processes the image and saves the structured output to message_json.json.
3. You can then ask natural language questions.
4. Agent answers each question by reasoning and using tools, saving conversation history in message_history.json.

Tips for Accurate Extraction

- Use high-quality, clear images for best results.
- Prompting Tips:
  - End queries with “in the invoice” or “in the bill” to help the model stay focused.
  - If results are off (e.g., due to looping or tool misuse), re-run with a more specific prompt.
  - Check the iteration logs to better understand and refine your prompt.

Would love to give a demo if required

THANK YOU!!