AIMS Assignment : Vision Language Models for Recipe Generation

1. Project Overview

Objective:
To generate concise 2–3 step cooking instructions from food images and noisy titles using Vision-Language Models (VLMs).

Approach:
Leveraged few-shot learning and prompt engineering with state-of-the-art VLMs to process multimodal data (images and text) and synthesize brief, actionable recipes.

---

2. Implementation Details

2.1 Model Selection

Primary Model: Gemini 2.0 Flash

- I chose this model because it offers fast inference, supports multimodal inputs (both images and text), and is easy to use. Additionally, it has been trained on a large, high-quality dataset, which contributes to its strong performance and reliable outputs.

Output Performance

- **BLEU Score: 0.0461** — Strong for open-ended tasks like food captioning, where exact word matches are rare.

- **ROUGE-1 Score: 0.4650** — Good Score even comparable to the fine-tuned LLaVA Chef model, indicating high-quality generation.

Method	Inputs	BLEU-1	BLEU-2	BLEU-3	BLEU-4	SacreBLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
Chef Transformer [16]	X_{ing}	0.267	0.127	0.064	0.034	0.038	0.116	0.262	0.059	0.136
Mistral [22]	$X_t + X_{ing}$	0.130	0.075	0.048	0.033	0.041	0.082	0.188	0.058	0.111
LLaMA [47]	$X_t + X_{ing}$	0.252	0.129	0.072	0.043	0.053	0.156	0.293	0.077	0.156
LLaVA [32]	$X_i + X_t + X_{ing}$	0.297	0.159	0.089	0.042	0.061	**0.2**	0.368	0.106	0.183
LLaVA-Chef-S1	$X_i + X_t + X_{ing}$	0.322	0.19	0.117	0.075	0.096	0.159	0.404	0.141	0.217
LLaVA-Chef-S2	$X_i + X_t + X_{ing}$	0.331	0.193	0.118	0.075	0.09	0.159	0.396	0.136	0.213
LLaVA-Chef-S3	$X_i + X_t + X_{ing}$	**0.362**	**0.215**	**0.135**	**0.089**	**0.167**	0.188	**0.473**	**0.172**	**0.241**

Source : LLaVA-Chef: A Multi-modal Generative Model for Food , Recipes arXiv:2408.16889v1 [cs.CL] 29 Aug 2024

Secondary Model: BLIP- 2

- A good Multi Modal Vision Language Model with most likes and most downloads on Hugging Face Platform.

Output Performance

- **BLEU Score: 0.0129** (limited success) — The Model had certain challenges like heavy size , computation difficulties and few errors which I could not resolve.

2.2 Dataset

- **Data Collection :** Manual Scraping of Images , Titles and Summary description from sources like Food.com , AllRecipes.com , Google Images etc.

- **Dataset Size :** Set of 10 curated samples for few shot training and 5 samples for test set.

  Below is an example of how the Recipe was stored :

Original Recipe

Japanese Egg Salad Sandwich

Image: Image of Japanese Egg Salad Sandwich

- **Ingredients**

- 4 large eggs
- ½ cup mayonnaise
- ¼ teaspoon kosher salt
- ¾ teaspoon white sugar
- 1 teaspoon Dijon mustard
- 3 dashes hot sauce
- 1 teaspoon lemon juice
- 2 teaspoons rice vinegar
- 1 pinch cayenne pepper
- 1 tablespoon heavy cream
- 4 slices soft white bread
- 1 tablespoon unsalted butter

- **Instructions**

- Steam eggs for 11 minutes, cool and peel
- Mix mayo, salt, sugar, mustard, hot sauce, lemon juice, and vinegar
- Mash eggs with seasonings and mayo
- Chill for 1 hour
- Butter bread and assemble sandwiches

→

Summarized Recipe


```
{
  "input": {
    "image": "image of sandwich",
    "title": "Egg Spread Sammy"
  },
  "output": {
    "summary": {
      "Ingredients": [
        "4 large eggs",
        "1/2 cup mayonnaise",
        "1/4 teaspoon kosher salt",
        "3/4 teaspoon white sugar",
        "1 teaspoon Dijon mustard",
        "3 dashes hot sauce, or to taste",
        "1 teaspoon freshly squeezed lemon juice",
        "2 teaspoons rice vinegar",
        "1/2 teaspoon kosher salt, or to taste",
        "1/4 teaspoon white sugar",
        "1 pinch cayenne pepper",
        "1 tablespoon heavy cream",
        "4 slices soft white bread",
        "1 tablespoon unsalted butter, softened"
      ],
      "Instructions": [
        "Steam eggs until hard boiled, then cool and peel",
        "Mix mayonnaise, salt, sugar, mustard, hot sauce, lemon
juice, and rice vinegar to make Kewpie-style mayo",
        "Mash eggs with seasonings, cream, and Kewpie mayo; chill
for at least 1 hour",
        "Butter bread slices, spread egg salad evenly, assemble
sandwiches, and optionally remove crusts"
      ]
    }
  }
}
```

2.3 Prompting Strategy

Two-Segment Prompting:

- **System Prompt:** A fixed description that defines the role, purpose, and expected behavior of the model. It sets the foundation which the model should follow throughout the task.
- **User Prompt:** The model is provided with Input-output pair examples for few-shot learning then it produces the output on test data in the same input format.

Overall , I used multiple prompting techniques like Chain of Thought (CoT) and Tree of Thought (ToT) where I provided the model with stepwise instructions like step 1 , step 2 etc along with looking for variations in the summary and , role-play where the model acts as an expert in culinary domain , this allows the model to behave like an expert and think effectively. This enables the model to enable structured, context-aware, and multi-step problem solving.

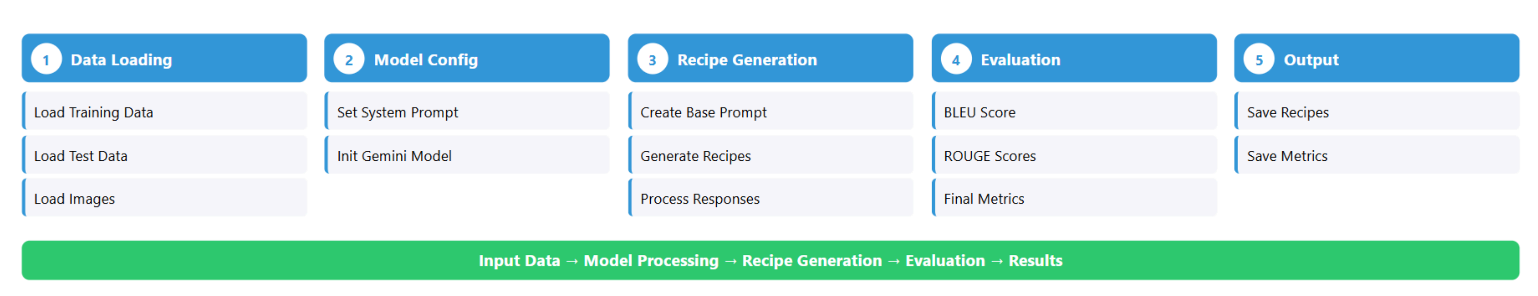
Template Structure:

- JSON-based prompt format for consistency and reusability.

2.4 Performance Metrics

- **BLEU Score:** 0.0461
- **ROUGE Scores:**
  - ROUGE-1 F1: 0.4650
  - ROUGE-2 F1: 0.1612
  - ROUGE-L F1: 0.2910

2.5 Pipeline



---

3. Challenges

3.1 Technical Challenges

- **Data Collection:**
  Manually curated samples from Food.com, AllRecipes, and Google and setting up the format. Focused on variety and quality. This approach took a little time and effort.

- **Multimodal Prompting:**
  Combined image and text inputs using structured prompts with few-shot examples.
- **Memory Constraints:**
  Faced issues with large models like BLIP-2. Had to optimize Python code and used Gemini 2.0 for efficiency.

3.2 Model-Specific Challenges

- **BLIP-2:** Encountered several implementation errors and computation problems.
- **Gemini 2.0:** Successfully integrated with consistent and meaningful output.

---

4. Future Improvements

- Improved prompt fine-tuning for better alignment and coherence.
- Expand the dataset for better generalization and training opportunities.
- Optimize computational backend for running heavier VLMs.
- Explore more advanced models like LLaVA-Chef for further accuracy.
