Multimodal Cooking Assistant

Crafting Recipes from Images and Titles

Vision Language Models

1. Overview

This research project explores the application of Vision-Language Models (VLMs) in generating concise cooking instructions from food images and noisy titles. Using a manually curated dataset of food items, a VLM-based pipeline was created to produce meaningful 2–3 step instructions for unseen dishes.

2. Objective

Generate short, clear cooking instructions from a food image and a vague title using a generalizable AI pipeline powered by a Vision-Language Model with the help of few-shot prompts.

3. Choice of Model (Gemma-3-4b-it)

- I had first thought of using **InternVL3-8B/2B/1B** and **Qwen-2.5VL** , which was working pretty well in Zero-Shot prompts but when it was required to give Few-Shot Prompts, the model was exceeding the memory limits and thus crashing.
- I then tried **BLIP2** but it was more of a Q/A model and was unable to follow the instructions given in the prompt and couldn't elaborate on things, thus wasn't suitable for summary generation.
- **LLaVa** wasn’t able to perform well on Few-Shot Prompts and **CLIP** was only for classification purposes and was not suitable for summary generation.
- In the end I settled for **Gemma-3-4b-it,** the latest Vision-Language Model released by Google just a few days ago. It provided great **Visual Feature Extraction Abilities** along with being **light-weight**, thus **easily handling Few-Shot Prompts** and **fine tuned for instruction** handling.

4. Dataset Setup

Web scraped 19 samples of recipes, each included:

- Noisy title (e.g., "spiced shred stack")
- Image file
- Full original recipe
- 2–3 step manually written summary

4.1 Example Entry (Train Set):

Title: *cheesy wheel*
Original Instruction (excerpt):

1. *Place a pizza stone on an outdoor grill directly over a wood fire. You may first need to spread out the wood if the flames are too high.*
2. *Roll out pizza dough on a flat surface to desired thickness. Place on pizza stone; cook until golden on one side, about 10 minutes.*
3. *Spread pizza sauce in an even layer onto cooked side dough, leaving 1/2-inch edge bare. Sprinkle mozzarella cheese on top of sauce; top with pepperoni.*
4. *Place on pizza stone; tent with aluminum foil and cook until cheese has melted, about 10 minutes. Transfer pizza to a cutting board. Cool slightly before cutting and serving.*

Manual Summary (Markdown Format):

*• **Prep Fire & Dough**: Place pizza stone on grill over wood fire. Roll dough to thickness.*
*• **Cook Pizza**: Cook one side 10 min, flip, spread sauce, cheese, and pepperoni. Tent with foil and cook until cheese melts (~10 min).*
*• **Serve Slices**: Cool slightly, slice, and serve.*

5. Workflow of the Model

1. **Web Scraping of Recipe Data:** Collected food images and corresponding detailed recipe instructions from online sources.
2. **Manual Annotation and Curation:** For each recipe I had created A **noisy or vague title** (e.g., "cheesy wheel", "creamy pink layer") and a **2–3 step concise summary** capturing the essential cooking process. This was done for 10 examples to construct a **few-shot prompt set**.
3. **Few-Shot Prompt Construction:** Formatted the 10 curated examples (title, image, and summary) into a structured prompt compatible with the instruction-following format expected by the Vision-Language Model.
4. **Model Inference (Using Gemma-3-4b-it):** Provided the model with a **new (unseen) food image and its noisy title**. The model generated a 2–3 step cooking summary based on the prompt structure learned from the few-shot examples.
5. **Prediction Evaluation (Using ROUGE Metrics):** Compared each model-generated summary with the **original manually written summary** (which was never shown to the model). Used **ROUGE-1**, **ROUGE-2**, and **ROUGE-L** to assess content overlap and structure similarity.

6. Analysis of the Results

6.1 When Model Performed Well

- The model consistently performed well in cases where the food image had **clear visual cues** and the **noisy title, though informal, still pointed toward a recognizable dish category**.
- Examples like "**charred sea pops**" for grilled shrimp or "**noodly tangle**" for pasta salads worked particularly well. In such instances, the model generated precise, 2–3 step summaries that **captured essential cooking actions** like boiling, sautéing, or grilling.
- Its effectiveness was largely due to the **consistency between visual features and the few-shot examples** used during prompting. The inclusion of **manually written summaries during setup helped the model internalize a structure that worked well** for common cooking workflows, especially when the steps were visually inferable and did not rely on hidden processes.

6.2 When Model Didn't Perform Well

- However, the model showed limitations in cases where the **image was ambiguous** or **compositionally complex**, such as **layered desserts or casseroles**, where the **internal steps (e.g., refrigeration, gelatin setting, or marination) were not apparent visually**.
- When the noisy title **lacked alignment** with the image or was **too abstract**—like "creamy pink layer"—the model occasionally hallucinated steps or produced vague, repetitive instructions. A common failure pattern was the **omission of temporally dependent steps** and a **tendency to favor simplified summaries** that fit the structure of the training examples, even if they weren't accurate.
- These errors **highlight the model's sensitivity to alignment between title, image, and training distribution**, and suggest that **improving visual-text grounding or diversifying prompt examples could mitigate such issues.**

7. ROUGE Score

```
====================== Aggregate ROUGE Scores ======================
                          ROUGE-1 : 0.508
                          ROUGE-2 : 0.147
                          ROUGE-L : 0.358

====================== Per-Recipe ROUGE Scores ======================

                Dads Creamy Cucumber Salad Recipe.jpg:
                            ROUGE-1: 0.513
                            ROUGE-2: 0.263
                            ROUGE-L: 0.410

               Bang Bang Chicken Casserole Recipe.jpg:
                            ROUGE-1: 0.531
                            ROUGE-2: 0.144
```

```
                    ROUGE-L: 0.389

    Strawberry Rhubarb Crumble Recipe.jpg:
                    ROUGE-1: 0.537
                    ROUGE-2: 0.113
                    ROUGE-L: 0.315

  Chef Johns Nashville Hot Chicken Recipe.jpg:
                    ROUGE-1: 0.408
                    ROUGE-2: 0.083
                    ROUGE-L: 0.306

  Classic Macaroni Salad Recipe with Video.jpg:

                    ROUGE-1: 0.543
                    ROUGE-2: 0.133
                    ROUGE-L: 0.370
```

To quantitatively evaluate the quality of the generated cooking summaries, I employed **ROUGE metrics**, commonly used for assessing text summarization tasks. Despite the constraints of a minimal dataset and vague multimodal inputs, the model achieved encouraging performance:

- **ROUGE-1 (0.508):** Indicates that over 50% of the key words in the reference summaries were successfully captured. This reflects **strong content retention** and suggests that the **model is adept at identifying core culinary actions** and ingredients from both the image and title.
- **ROUGE-2 (0.147):** As expected in abstractive generation tasks, this score confirms that the model **still manages to reconstruct meaningful short phrases** and demonstrates a **grasp of cooking step sequences**, especially for dishes with clear visual cues.
- **ROUGE-L (0.358):** The relatively high ROUGE-L score shows that the model was able to **preserve the logical ordering** and **structural coherence of instructions**, even when abstracting from complex recipe content. This reinforces the model's strength in **producing readable, logically structured outputs**.

On a per-recipe level, the model **performed best when the dish's visual presentation clearly corresponded with recognizable cooking patterns** (e.g., salads, grilled items, or pasta-based dishes), achieving **ROUGE-1 scores as high as 0.543**. In more visually ambiguous cases, such as layered desserts or marinated dishes, the scores dipped slightly—highlighting the model's reliance on visually inferable cues rather than implicit processes.

Importantly, these results **validate the effectiveness of few-shot prompting** using just 10 manually curated examples. The model consistently produced summaries that maintained **high**

lexical overlap and structural fidelity to ground-truth references, even without any model fine-tuning. This affirms the core hypothesis of the project: that general-purpose VLMs can be adapted for domain-specific tasks through lightweight, instruction-tuned prompting strategies.

8. Novelty of Solution

- A central novelty of this project lies in its **few-shot instruction** generation using manually curated summaries, which bypasses the need for traditional finetuning.
- Instead of training on large datasets, the model was guided using just **10 carefully crafted examples** of 2–3 step cooking instructions derived from full recipes. This approach strikes an effective balance between data efficiency and adaptability, proving that meaningful, generalizable outputs can be produced even with limited supervision.
- Additionally, the solution features a **manual alignment of noisy titles with visual context**, **avoiding the complexity of full image captioning** or detection pipelines. By conditioning the model on minimal image features and a vague title, the method simplifies the multimodal input while still achieving relevant and coherent instruction generation.
- This streamlined design showcases how lightweight prompting strategies can unlock strong performance from compact models like **Gemma-3 4B,** making the solution both practical and replicable on resource-constrained systems.

9. Problems Faced and Learnings

1. **Model Selection and Prompt Compatibility**
    - **Challenge:** Initially, multiple Vision-Language Models (InternVL, Qwen-VL, BLIP2, LLaVa) were explored. Although some performed well under zero-shot settings, they consistently failed under few-shot prompting due to **memory crashes** (InternVL/Qwen) or poor instruction-following capabilities (BLIP2, LLaVa).
    - **Learning:** Model choice isn't just about accuracy—it also depends heavily on **hardware constraints and prompt adaptability**. Lighter models like Gemma-3-4b-it proved more practical, showing that well-instructed small models can outperform heavier ones if used efficiently.

2. **Dealing with Noisy and Ambiguous Inputs**
    - **Challenge:** Noisy titles like “creamy pink layer” or “zesty twist bowl” often **lacked clear semantics or context**. When paired with complex or abstract food images (e.g., layered desserts), the model sometimes hallucinated steps or skipped important processes like refrigeration or marination.
    - **Learning:** The **alignment between the title, image, and prompt** examples is crucial. Clearer titles or preprocessed semantic hints could improve grounding.

This also highlighted the need for visual reasoning capabilities in models for tasks beyond surface-level inference.

3. **Few-Shot Prompt Engineering**
    - **Challenge:** Designing effective few-shot prompts that guide the model toward structured, concise outputs without overwhelming it was non-trivial. Including **too many examples caused prompt overflow; too few led to underperformance**.
    - **Learning:** Few-shot learning benefits from **quality over quantity**. A small set of well-structured, diverse prompts can teach the model general patterns. The process also emphasized the importance of prompt formatting consistency to ensure the model generalizes well across inputs.

4. **Data Collection and Curation**
    - **Challenge:** Manually curating datasets—especially creating accurate 2–3 step summaries from full recipes—was **time-consuming**. Ensuring that each summary preserved core cooking actions without oversimplifying was a fine balance.
    - **Learning:** Hands-on data curation sharpened my **understanding of instructional abstraction**—how to reduce a multi-step process into its essence while keeping it informative. It also reinforced the value of domain knowledge in building effective VLM prompts for specific use-cases.

5. **Model Generalization vs Specificity**
    - **Challenge:** The model sometimes overly mimicked the structure or style of training examples, even when the dish context changed. This made it less adaptable to visually or semantically novel inputs.
    - **Learning:** Instruction-following VLMs can develop a rigid response template if prompt examples aren't varied. A more diverse training set—even with just a few more varied examples—can help models better generalize.

10. Video Walkthrough of The Project:

https://drive.google.com/drive/folders/1aHI_y5v9_6sz440fqQDc_02UCFv_1vie?usp=sharing