SafePlates: An NLP Approach to Allergy-Safe Recipe Modification and Evaluation

Chang Su
Dept. of Electrical Computer Engineering
University of Toronto
cs.su@mail.utoronto.ca

Xiaoyang (Jason) Xie
Dept. of Electrical Computer Engineering
University of Toronto
xiaoyang.xie@mail.utoronto.ca

Linxin Li
Dept. of Mechanical Industrial Engineering
University of Toronto
linxin.li@mail.utoronto.ca

Tianli Xu
Dept. of Mechanical Industrial Engineering
University of Toronto
tianli.xu@mail.utoronto.ca

Final Report
Word count: 1991
Penalty: 0%

ECE 1786: Creative Applications of Natural Language Processing (Fall 2024).

Permissions

Permission to post video	Yes
Permission to post final report	Yes
Permission to post source code	Yes

1 Introduction

This project aims to address the challenge of accommodating dietary restrictions and preferences by leveraging large language models (LLMs) to create recipes that substitute allergenic or restricted ingredients with suitable alternatives. By intelligently recommending substitutions, the system ensures safe, diverse, and satisfying meal options while preserving the integrity of original recipes. Our goal is to enhance the culinary experience for individuals with dietary needs, making enjoyable and accessible meals a reality for all.

2 Illustration

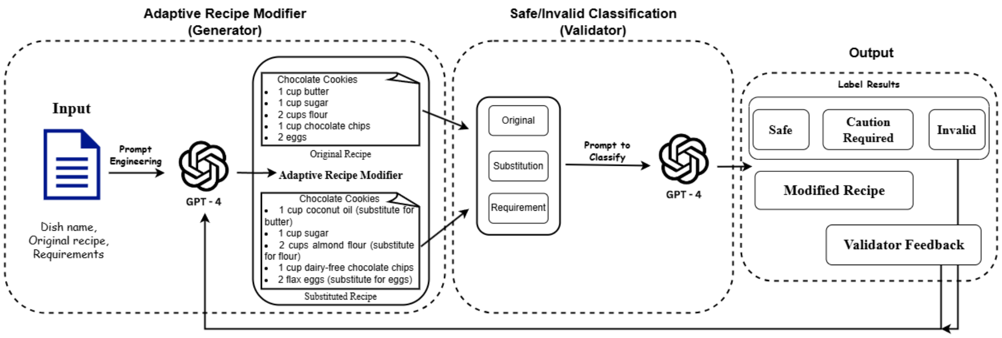

Figure 1: Architecture

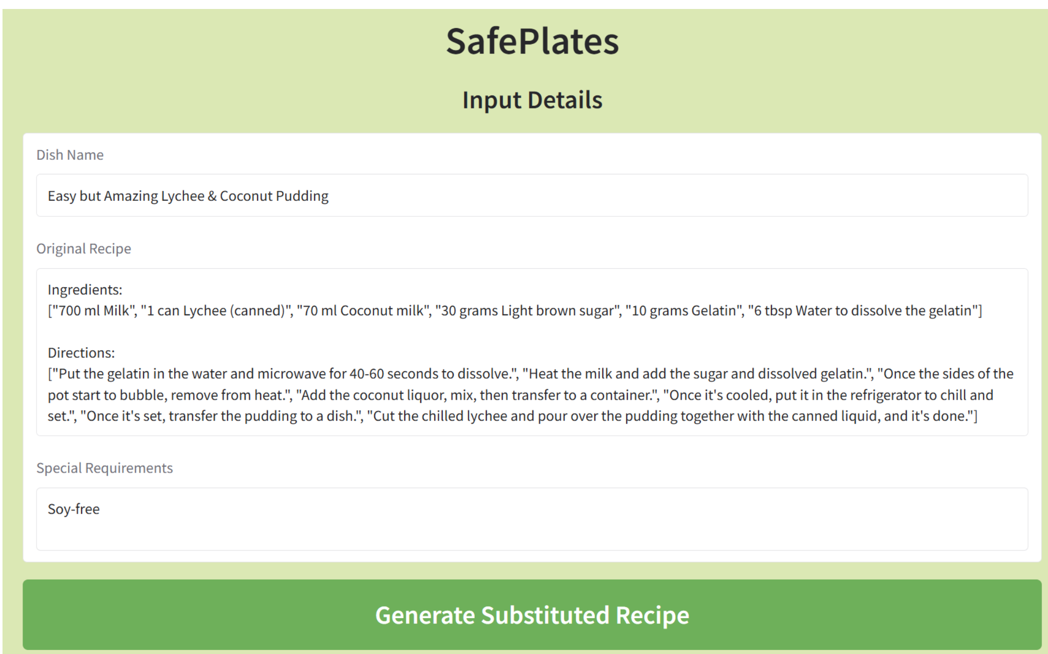

Figure 2: Application Input Example

3 Background & Related Work

In this section, we are going to discuss all the research we conducted to build this application.

1. The research paper Integrating Expertise in LLMs: Crafting a Customized Nutrition Assistant with Refined Template Instructions (1) explores how prompt engineering improves the

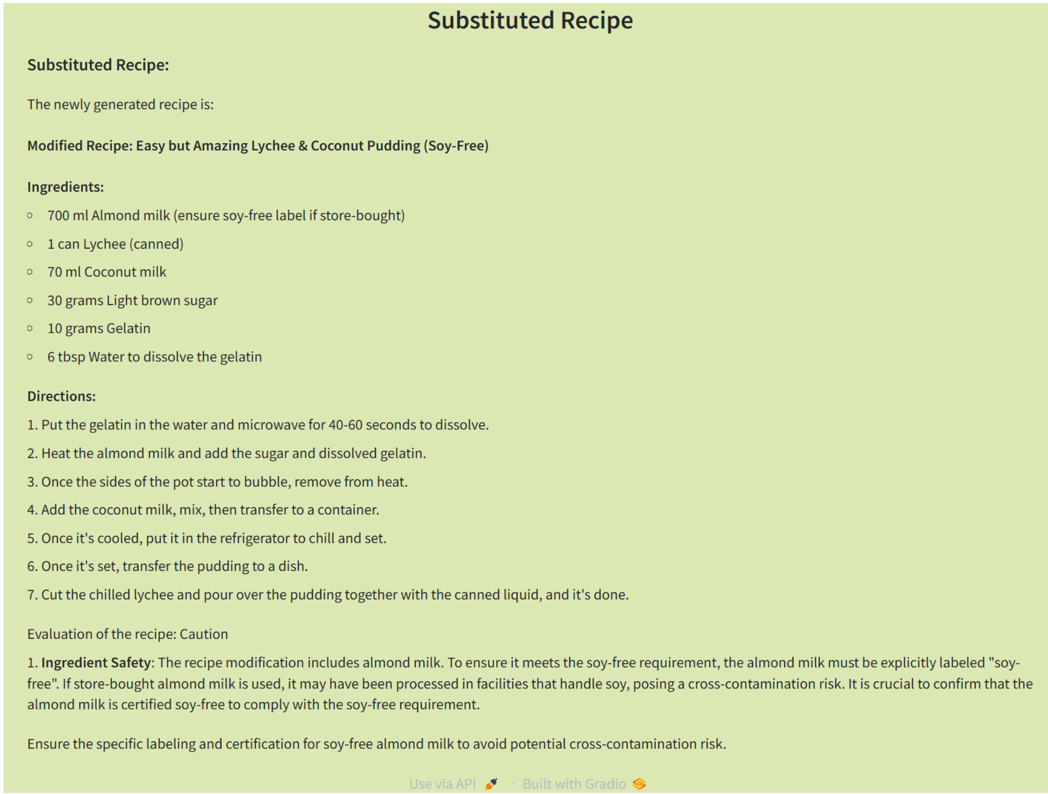

Figure 3: Application Output Example

accuracy and personalization of food and nutrition information generated by LLMs, offering insights and suggestions for structuring effective prompt templates.

2. A prompt pattern catalog (2) informs our approach to prompt engineering in SafePlate by introducing structured patterns. Techniques include a **sequenced instruction pattern** for coherent, step-by-step recipes and a **persona-driven prompt** that frames the LLM as a culinary expert, supporting reliable ingredient substitutions and dietary adjustments. **Fact-checking prompts** and **template structures** enhance consistency and safety, ensuring user-centered recipe outputs.

4 Architecture and Software

1. **Input:**
   - Users provide the name of the dish, the original recipe, and any relevant dietary requirements.(see Figure 4 below)
2. **Generator:**
   - The **Generator** module produces a modified recipe tailored to the specified dietary restrictions. This process begins by integrating the user's provided input (dish name, original recipe, and dietary requirements) into a structured prompt.(see Figure 4)
   - The prompt is then processed by the GPT-4o model, which generates a revised recipe. If this represents a subsequent iteration (detected not the first attempt to generate this recipe), additional feedback from the **Validator** module (see the Validator section below) is collected and incorporated into the prompt to re-generate a new recipe.

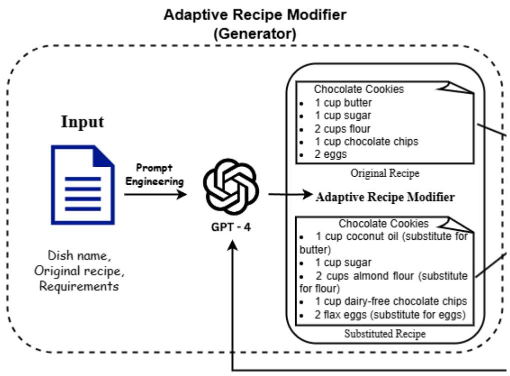

Figure 4: System Overview of the Generator Module

3. **Validator:**
   - The **Validator** module evaluates the modified recipe, comparing it against both the original recipe and the user-specified dietary restrictions. Its primary goals are to ensure culinary feasibility, taste, and adherence to safety and dietary guidelines.
   - The **Validator** The system classifies the revised recipe into one of the following categories **Safe**, **Caution Required**, and **Invalid**. In Section 5.2, we provide detailed definitions for these three categories.
   - The **Validator** will provide feedback with explanations and instructions for the **Caution Required** and **Invalid** categories.

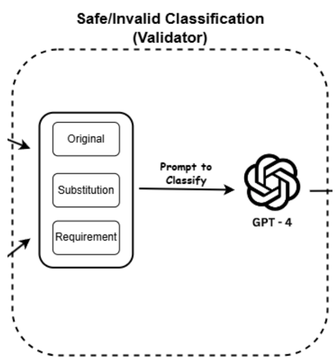

Figure 5: System Overview of the Validator Module

4. **Output to Users:**
   - **Safe**: If the recipe is classified as safe, the generated recipe is presented directly to the user.
   - **Caution Required**: If the Validator categorizes the recipe as requiring caution, the recipe is presented to the user along with the Validator's advisory feedback.
   - **Invalid**: If the recipe is deemed invalid, the feedback provided by the Validator will be collected and add to the prompt and reinitiate the generation process, with more information and instructions provided, aiming to produce a revised recipe that meets the specified requirements.

5 Data and Data Processing

RecipeNLG (3), a large dataset containing over one million recipes, we use this dataset a starting point to create our own dataset.

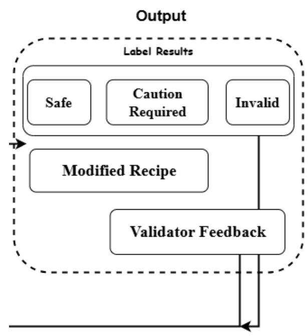

Figure 6: Output to Users

5.1 Availability Dataset

The purpose of creating the availability dataset is to evaluate the system's capability to generate results based on real-world recipes while accommodating a diverse range of user requests. To achieve this, 800 recipes were randomly selected from RecipeNLG. Among these, 400 recipes were tailored to include specific dietary restrictions, such as "Gluten-free," "Dairy-free," or "Peanut-free." An additional 200 recipes were adjusted to reflect dietary preferences, such as "I am trying to lose weight," "Make it sugar-free," or "Replace all meat with pork." The remaining 200 recipes were adapted to meet requirements for specific groups of people, ensuring comprehensive coverage of various user needs.

5.2 Validation Dataset

The validation dataset is designed to evaluate the functionality and performance of the validator agent. Its purpose is to ensure the agent accurately identifies incorrect or unsafe recipes generated by the generator and provides actionable feedback to improve it. The dataset consists of 52 entries spanning three categories: safe, caution, and invalid, with scenarios comprehensively covered for each category.

Definitions: What defines a valid recipe.

- **Edibility:** The modified recipe must remain edible.
- **Requirement Satisfaction:** The modified recipe must fully meet the specified requirement or objective.
- **Preservation of Core Characteristics:**
  - The modified recipe should maintain the fundamental essence of the original dish (e.g., key flavors, textures, or techniques).
  - Deviations should align with the intent of the requirement and not compromise the identity of the dish.
- **Logical Cooking Steps:** All cooking steps must be clear, coherent, and executable without contradiction or omission.
- **Practicality:** The modified recipe should be achievable with reasonable effort, tools, and commonly available ingredients.
- **Avoidance of Hazard:** The recipe must not include any unsafe practices, harmful combinations, or ingredients that could pose health risks.

Based on the definition of a valid recipe we can define the 3 categories as:

- **Safe:** The generated recipe satisfy all property defined above.
- **Caution:** The generated recipe satisfy all property defined above in general, but has subtle issue with requirement fulfillment and avoidance of hazard due to the nature of the recipe:

1. Allergen Cross-Reactivity: The recipe includes ingredients that may cause cross-reactivity in individuals with specific allergies. (i.e., someone allergic to latex might also react to foods like bananas or avocados.)
2. Ambiguous Ingredient Names: The recipe uses unclear ingredient names that could potentially violate the requirements. (i.e., in a nut-free recipe, specifying "butter" instead of "nut-free butter.")
3. Excessive Use of Potentially Harmful Ingredients: The recipe contains ingredients that are safe in small amounts but could become harmful if used in excessive quantities. (i.e., Spirulina)

- **Invalid:** The generated recipe fails in one or more of the property defined above.

6 Comparison

1. **Evaluation of the Generator:**
   - The performance of the **Generator** will be assessed through extensive testing aligned with the criteria established by the **Validator**.
   - A diverse set of data will be employed to ensure that the **Generator** is tested under various conditions and scenarios.
   - The evaluation process will utilize the *Availability Dataset* to verify the following:
     - The Generator is capable of producing high-quality recipes in most cases.
     - The Generator meets a wide range of user dietary requirements while maintaining the integrity of the original dish.
2. **Evaluation of the Validator:**
   - The performance of the **Validator** will be assessed using the *Validation Dataset*.
   - Manual evaluation will supplement automated testing to verify the Validator's ability to handle complex and edge cases effectively.
   - The evaluation will focus on the Validator's ability to:
     - Correctly classify recipes according to their adherence to dietary requirements.
     - Accurately identify and explain issues in challenging scenarios, such as subtle violations of restrictions or deviations from expected culinary standards.

7 Quantitative Results

Category	**Safe**	**Caution**	**Invalid**	**Total**
Number of Tests	20	13	19	52
Mislabel to Safe	NA	1	0	1
Mislabel to Caution	0	NA	5	5
Mislabel to Invalid	0	2	NA	2
Number of Errors	0	3	5	8
Correct Rate (%)	100.0%	76.9%	73.7%	84.6%

Table 1: Validator functionality test

Category	**Allergy Free**	**Substitute Ingredients**	**Special Requirements**	**Total**
Number of Tests	400	200	200	800
Number of Errors	0	9	0	9
Correct Rate (%)	100.0%	95.5%	100.0%	98.9%

Table 2: Availability test

8 Qualitative Results

- **Validator Functionality Test:**

  - **Safe Recipes:** Achieved 100% accuracy, demonstrating robust performance in accurately identifying safe recipes without error.
  - **Caution and Invalid Recipes:** Achieved accuracy rates of 76.9% and 73.7%, respectively, reflecting challenges in detecting subtle hazards or dangers stemming from complex chemical reactions. These issues may arise due to limitations in the model's deep logical inference capabilities or the need for further prompt optimization.
  - **Overall:** Achieved an 84.6% correct rate, with no extreme false positives or false negatives, indicating solid functionality while highlighting opportunities for improvement in handling edge cases.

- **Availability Test:**
  - **Allergy-Free and Special Requirements:** Maintained 100% availability, demonstrating excellent adaptability to allergy-related needs and specific requirements.
  - **Substitute Ingredients:** Achieved 95.5% availability. Rejected cases fall into two categories: (1) overlooking an ingredient in the original recipe and making an incorrect substitution, and (2) a conflict between the requirement and the original recipe, resulting in no valid solution. For the first type of failure, further refinement of the prompt may be necessary.
  - **Serve for Certain Groups of People:** Maintained 100% availability, demonstrating excellent adaptability in handling such cases. However, it is expected that some failure cases may occur in practice, as certain recipes may inherently remain unsuitable for specific groups regardless of modifications.
  - **Overall:** Delivered a 98.9% available rate, reflecting high reliability in meeting customization requirements with minimal errors.

9 Discussion and Learnings

The project results demonstrate notable success, with an 84.6% accuracy in the validator functionality test and a 98.9% success rate in providing valid substitutions within two attempts. These outcomes highlight the system's reliability in addressing common dietary needs, such as creating allergy-free recipes or modifying ingredients based on user requirements. The validator's key strength lies in ensuring no "Caution" or "Invalid" cases were misclassified as "Safe," reinforcing the safety and trustworthiness of the system.

Despite these achievements, certain areas revealed opportunities for improvement. The validator often struggled with complex edge cases, particularly for "unsafe" and "caution" recipes, where ingredient chemical properties changed during the cooking process or where ingredient quality was compromised; For example, it uses Aspartame as sugar substitute in cookie, but Aspartame will decomposes at cooking temperatures, which make the recipe not safe.

Additionally, while the model effectively preserves flavor profiles in most scenarios, some substitutions failed to maintain the original taste integrity, indicating a need for improved culinary precision. For instance, we included several test cases with requirements such as "less spicy" or "less sugar." However, our agents often modified ingredient quantities inappropriately, either reducing them excessively or adding too much. The validator is not sensitive to changes like this.

This project underscored the importance of combining AI innovation with domain expertise. Success relied on well-structured datasets and robust validation processes, but challenges like maintaining flavor integrity and addressing edge cases highlighted the need for deeper understanding of ingredient properties and culinary principles.

In future projects, we would prioritize early collaboration with domain experts to design comprehensive validation datasets. Expanding datasets to cover diverse substitutions and integrating dynamic user feedback would improve adaptability. Finally, exploring advanced flavor mapping techniques and better ingredient interaction modeling would enhance the system's ability to meet dietary needs while preserving taste.

10 Individual Contributions

Jason Xie contributed extensively to the development and testing of the SafePlates project. He was responsible for implementing the generator module, including creating partial test cases to ensure its functionality. Jason also developed a new web-based UI using Gradio, merged the generator and validator components for a seamless workflow, and introduced modifications to the agent workflow between generator and validator to allow retries after validation. Additionally, he tested the system using availability dataset 2, and made adjustments to print out and save the results in order to optimize its performance. Jason further enhanced the project by adding detailed documentation through a README file, refining the Gradio-based UI, and deleting unused files to streamline the repository.

Tom Xu contributed significantly to the project by drafting the initial workflow and creating a Gradio demonstration in Jupyter Notebook as a reference. My focus was on enhancing the Generator through prompt engineering, refining it iteratively based on test results, and restructuring its input for a multiple-attempts approach. He also refined the Validation and Availability Datasets through manual sampling and adjustments, specially focusing on edging case with special requirements to ensuring diversity, relevance, and robustness to address edge cases and special user requirements.

Linxin Li contributed extensively to the SafePlates project by collecting and organizing datasets, including creating a dataset for the validator and adding recipes for caution categories. He implemented key testing functions, refined validator prompts for improved accuracy, and designed multiple test cases. He enhanced the UI and web display formats to ensure a cleaner, user-friendly interface. Additionally, He streamlined the folder structure and updated file naming conventions to improve project organization. He was also responsible for testing the availability features, merging and debugging test cases, and iterating on functionality to enhance the overall project quality and user experience.

Chang Su contributed to the project by designing both the initial two-category prompt and the subsequent three-category prompt for the validator agent. He proposed and clarified the boundaries for the three categories utilized in the validator, ensuring precise and actionable definitions. Additionally, Chang developed the standards and delineated the scenarios required to comprehensively cover manual test cases for the validator. He also contributed to the creation of portions of the validation and availability datasets. Furthermore, He performed and manually inspected the test results for both datasets to ensure accuracy and adherence to the project's requirements.

References

[1] Szymanski, A., Wimer, B. L., Anuyah, O., Eicher-Miller, H. A., & Metoyer, R. A. (2024, May). Integrating Expertise in LLMs: Crafting a Customized Nutrition Assistant with Refined Template Instructions. In Proceedings of the CHI Conference on Human Factors in Computing Systems (pp. 1-22). Retrieved October 31, 2024, from `https://dl.acm.org/doi/10.1145/3613904.3641924`.

[2] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, et al., "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT," *arXiv preprint arXiv:2302.11382*, 2023.

[3] Mooney, P. T. (2020, September). *RecipeNLG, Version 1*. Retrieved October 31, 2024, from `https://www.kaggle.com/datasets/paultimothymooney/recipenlg`.