Leveraging AI for Personalized Ingredient-Specific Recipe Generation

Lara Bernard[1], Sabrina Chen[1], Sophia Shi[1], Lynn Ye[1],
larab@mit.edu, sabchen@mit.edu, sophia_s@mit.edu, lynnye27@mit.edu

[1] Massachusetts Institute of Technology

Abstract

Household food waste is a large contributor to the global food waste crisis, which furthers the issues of world hunger, economic losses, and greenhouse gas emissions. We address this by developing an AI-powered recipe generation model that takes in a user's available ingredients and recipe preferences, including dietary restrictions and cuisine. The generated recipe uses ingredients users may need need to get rid and minimizes the need to buy additional ingredients that could go unused, all while following their preferences. We trained language generation models BART and LLaMA3 on a dataset of online recipes from Food.com. Our LLaMA3 model was then fine-tuned by implementing constrained decoding to generate recipes in a standardized format, penalize recipes that violate dietary restrictions, and reward those that maintain the authenticity of their cuisines. These models were evaluated based on ingredient accuracy, adherence to user preferences, and expected quality. From our evaluation, we found that the fine-tuned LLaMA3 model was best for generating recipes. Future work aims to further expand the authenticity of non-Western cuisines with improvements on our constrained decoding and evaluation. By prioritizing user requirements and ensuring cultural and dietary inclusivity, our model is a promising step toward mitigating food waste.

1 Introduction

As our society increasingly aligns with international consumerism, food waste prevails as a global challenge. Every year, 1.03 billion tons of food waste are generated, which constitute 17% of global food production (Li, et al., 2023). This contributes to approximately 3.3 billion tons of greenhouse gasses and $750 billion of economic loss. Households generate approximately half of this waste. For example, about 70% of food waste in the UK in 2018 came from households, two-thirds edible, because of a lack of knowledge about food edibility and recipes (Teng et al., 2021). However, as the World Health Organization reported in July 2024, about 1 in 11 people, or 733 million worldwide, faced hunger last year. This issue of food insecurity and malnourishment can be mitigated using the millions of metric tons of food wasted every year. Thus, food waste and its consequential impacts on the climate, economy, and food security are multi-pronged issues that can be addressed with food and leftover management strategies.

Cooking at home is a cornerstone of daily life, with 81% of consumers preparing more than half of their meals at home, according to the National Association of Frozen and Refrigerated Foods. Of these consumers, 89% use digital recipes (Petrak, 2023). However, it can be difficult to find recipes that align with ingredients they have on hand. This often means extra effort to shop for specific items, as 75% of consumers use recipes to guide their grocery choices (Petrak, 2023). Additionally, dietary preferences like taste, which is considered by 66% of consumers, and healthiness, 49%, further complicate the decision-making process (Mitchell, 2023). Finding the right recipe can make cooking at home feel daunting and discourage people. Yet, cooking at home offers significant benefits: it's more affordable, healthier and better for the environment by reducing energy use and resource consumption (Hoffman, 2012). Addressing these barriers could empower more people to enjoy cooking at home.

To simplify the path from ingredient to recipe and mitigate food waste, we propose a refined recipe generation model based on available ingredients and desired user preferences. We test different large language models and fine-tuning strategies to determine which one is the best at producing recipes for each cuisine. To evaluate our model, we assess ingredient accuracy, recipe preferences, and expected rating, three important metrics for an effective recipe. Ultimately, we have a final recipe generator which uses whichever model is best to generate a recipe for the user.

2 Related Work

Language models have shown promising results in various areas, from speech recognition and language translation to applications in climate sciences, finance, and other data-driven solutions. Recent AI advancements in the culinary field have prompted the development of new models and systems capable of generating recipes, suggesting ingredients, and even classifying images of different dishes and cuisines. For example, researchers have created RecipeGPT, a model that generates recipes based on ingredient input (Lee, et al., 2020), and applied computer vision in food image classification and nutrient identification (Kaur, et al., 2023). Despite these advancements, most of these methods are limited in scope of inputs and lack robust evaluations of the effectiveness of generated recipes. Instead, researchers focus on coherence and quality of instructions at the expense of generalizability, customization, and user satisfaction.

Current recipe generators do not address a user's cuisine and ingredient preference. In Majumder et al. (2019), they use an encoder and two-layer GRU decoder with a user's historical taste preferences as input to generate a new recipe. Though this model uses historical user data to generate new recipes based on those preferences, allowing personalization, it does not assess how well the generated recipes match specific cuisines or how the given ingredients are efficiently utilized. In addition, in Aljbawi (2020), recipes are generated using three different models. The first is association rules, which looks for similarities between subsets to pick some combination of ingredients. The second is deep neural networks, which uses patterns to pick ingredients that normally go together. The third is BERT, which uses NLP to predict which ingredients are most likely to follow each other. However, this research similarly does not address input cuisines or ingredients to use provided by the user, despite assessing intra-ingredient compatibility and coherence.

With current recipe-generating models, there is a discrepancy between the recipes generated for Western-flavored dishes compared to non-Western-flavored dishes due to the model training being done on English recipes (Saxena, 2023). This means that when producing dishes of non-Western cuisine, they often taste worse and lack authenticity compared to dishes of Western cuisine. In Lee et al. (2020), the researchers evaluate cultural accuracy and creativity for generated recipes of different cuisines, they find that some cuisines lack variety across the different recipes generated for that specific cuisine. We circumvent this issue by fine-tuning our model so that it will specifically generate recipes of comparable value across all cuisines.

Existing models like GPT-3 often generate recipes that include unnecessary or unfamiliar ingredients, which can be impractical for users (Krishna and Metz, 2022). Moreover, these models sometimes overlook dietary restrictions. To address these limitations, our model takes a user-centered approach by prioritizing the ingredients users already have or prefer. This naturally aligns with their dietary restrictions since users input ingredients that reflect their preferences and needs (e.g., a vegetarian user wouldn't include beef as an option). Additionally, users can explicitly specify dietary constraints, such as vegan, gluten-free, or low-sodium, ensuring that the generated recipes are accessible and user-specific. This approach minimizes unfamiliar ingredients while maximizing utility.

Finally, frequent concern with AI-generated recipes is that they can be too generic, making them lack flavor (Saxena, 2023). In order to prevent our generation from recommending bland or bad-tasting recipes, we evaluate recipes based on how tasty we expect them to be. We do this by using similar existing recipes and their ratings to predict what the rating/user satisfaction with the generated recipe would be, our expected rating metric.

3 Methodology

3.1 Dataset

Our model is trained and tested on the "Food.com Recipes and Interactions" open-source dataset, a collection of over 230,000 online recipes (Li, 2019). Each recipe has accompanying information including an ingredient list, cuisine type, nutritional information (e.g., high-protein, low-fat), dietary restrictions (e.g., vegan, diabetic), meal type (e.g., breakfast, dessert), rating, and step-by-step instructions. This dataset is more suitable than others because it 1) has an extensive and diverse range of ingredients, preparation steps, and cuisine types (e.g., Polynesian, African, Lebanese), and 2) includes ratings, dietary restrictions, and nutritional information which enable the model to better accommodate a wide range of user preferences. Data preprocessing is conducted using the Pandas and NumPy libraries to extract relevant features and convert the dataset into easily manageable formats. The dataset is divided into a training set (70%) and a testing set (30%). The testing split is utilized to generate prompts formatted for recipe generation in the following structure:

```
Ingredients Available: [ingredients]
Preferences:
- Meal Type: [meal type]
- Dietary Restrictions: [dietary restrict.]
- Nutrition: [nutrition requirements]
- Cuisine: [cuisine]
Generate a recipe.
```

3.2 Tokenization and Model Fine-Tuning

For this study, we fine-tuned two state-of-the-art text generation models, LLaMA3 and BART, leveraging the Hugging Face Transformers library. The tokenization process used the respective tokenizers of each model for precise segmentation and encoding of input text, optimizing it for the fine-tuning process. Fine-tuning was conducted using a causal language modeling objective for LLaMA3 and a sequence-to-sequence objective for BART. LLaMA3, as a decoder-only model, generates recipes autoregressively by predicting the next token based solely on prior context, whereas BART, an encoder-decoder model, first encodes input prompts into latent representations before decoding them into recipe outputs. While both models showed strong capabilities, LLaMA3 was chosen for its superior performance and its compatibility with constrained decoding. The autoregressive nature of LLaMA3's architecture simplifies the integration of constraints during token generation, whereas BART's encoder-decoder framework introduces additional complexity due to the interplay between the encoder and decoder components. All computations for this project were performed using Google Colab's A100 GPU for efficient processing and model training.

3.3 Constrained Decoding

To enhance the alignment of the language model's outputs with user-defined requirements, we implemented constrained decoding via a custom logits processor. This approach introduces three targeted constraints: a formatting constraint, a dietary restriction constraint, and a cuisine relevance constraint. These constraints collaboratively guide the decoding process by adjusting token probabilities at each step, ensuring the generated recipes adhere to structural and dietary expectations.

3.3.1 Formatting Constraint

The formatting constraint enforces a standardized recipe structure, with an Ingredients section followed by a Steps section. The logits processor dynamically adjusts token probabilities to align with the current section. During the Ingredients section, tokens associated with the steps are penalized. Similarly, during the Steps section, tokens associated with the ingredients are penalized.

3.3.2 Dietary Restriction Constraint

The dietary restriction constraint enforces adherence to user-defined dietary preferences by penalizing tokens associated with ingredients that violate these restrictions. To do this, we utilized ChatGPT-4 to generate a comprehensive list of 50 words that represent violations for each dietary restriction (e.g., animal-derived ingredients such as chicken and cheese for vegan recipes, or gluten-containing items like wheat and rye for gluten-free recipes). These lists serve as the basis for identifying tokens to penalize during decoding. For each restricted token, a penalty weight of -4.0 is applied to its logits, significantly reducing the likelihood of selection while maintaining the model's ability to generate coherent and diverse recipes. The penalty weight was empirically determined to achieve a balance between strict adherence to dietary constraints and flexibility in token selection. This approach ensures that the generated recipes align with user dietary requirements without compromising its fluency or contextual relevance.

3.3.3 Cuisine Relevance Constraint

The cuisine relevance constraint ensures the generated recipes align with the requested culinary style by boosting the likelihood of tokens associated with the specified cuisine. Similar to the dietary restriction constraint, we leveraged ChatGPT-4 to generate a list of 50 cuisine-adjacent words for each cuisine (e.g., basil and parmesan for Italian, or soy sauce and ginger for Chinese). For each token associated with the target cuisine, a boost of +1.5 is applied to its logits, increasing the likelihood of its selection during token generation. The absolute weight of +1.5 was intentionally chosen to be lower than the -4.0 penalty applied for dietary restriction violations because of the higher importance of dietary adherence compared to cuisine alignment. This approach ensures that generated recipes incorporate culturally relevant ingredients while balancing adherence to dietary constraints.

4 Evaluation

We evaluated the generated recipes using three primary metrics: ingredient accuracy, recipe preferences, and expected rating. The three scores generated from these metrics are aggregated into an overall score for that recipe. We average the overall scores for the 69,492 recipes in the testing dataset to evaluate how well that model and the addition of fine-tuning perform. This comparative evaluation will allow us to analyze qualitative differences among models and identify which recipe generator performs best.

4.1 Ingredient Accuracy

Ingredient Accuracy is calculated as an average of the following two scores.

4.1.1 Input Ingredient Accuracy

Input Ingredient Accuracy is calculated by looking at the percentage of user-provided ingredients that are accurately retained in the output. This metric helps mitigate food waste by ensuring that recipe generation prioritizes the use of user-specified ingredients, reducing the likelihood of discarding unused ingredients. The score is calculated as:

$$\frac{\text{\# of input ingredients in output recipe}}{\text{total \# of input ingredients}} * 100\%$$

4.1.2 Output Ingredient Accuracy

Output Ingredient Accuracy is calculated by looking at the percentage of ingredients from the generated recipe that are not included in the input ingredients list. We want to minimize the number of new ingredients the user must purchase for the recipe to prevent additional, unwarranted food waste.

$$\frac{\text{\# of output ingredients not in input}}{\text{total \# of output ingredients}} * 100\%$$

4.2 Recipe Preferences

This metric assesses how well the generated recipe adheres to the following four user-specific preference categories: cuisine (e.g., Indian, Egyptian), nutrition (e.g., high-protein, low-cholesterol), dietary restrictions (e.g., vegan, diabetic), and meal type (e.g., breakfast, dessert). For each generated recipe, we calculate the cosine similarity between the model's output and existing recipes based on ingredients and instructions. We use this to identify the 20 recipes most similar to it from our dataset. For each preference category, we evaluate the percentage of the 20 most similar recipes that adhere to the user preference. The final recipe preferences score will be calculated as an average of these four preference categories.

4.3 Expected Rating

Expected rating is calculated using the ratings of recipes in the dataset that are most similar to the generated recipe. We use the same method as we used for recipe preferences to find the top-five most similar recipes. We use the ratings of these to calculate a weighted average as a proxy for the generated recipe's expected rating. The score will be computed using a weighted similarity formula:

$$\sum_{i=1}^{5} \left(\frac{\text{similarity}_i}{\sum_{j=1}^{5} \text{similarity}_j} \times \text{rating}_i \right)$$

4.4 Overall Weighted Score

The final overall score is calculated as a weighted average of the previous 3 metrics: ingredient accuracy, recipe preference, and expected rating. We assign a 45% weight to ingredient accuracy, a 35% weight to recipe preferences, and a 20% weight to expected rating. Given our initial motivation and goal of mitigating food waste, we wanted to emphasize ingredient accuracy and recipe preferences. We assigned a low weighting to expected rating because recipe ratings are subjective and are often bimodal.

5 Results

5.1 Performance Metrics Results and Findings

BART and LlaMA3 generated recipes of certain formats. Below, we provide examples of the recipes that were generated.

These are the performance metric evaluation results of the three models that we tested: BART,

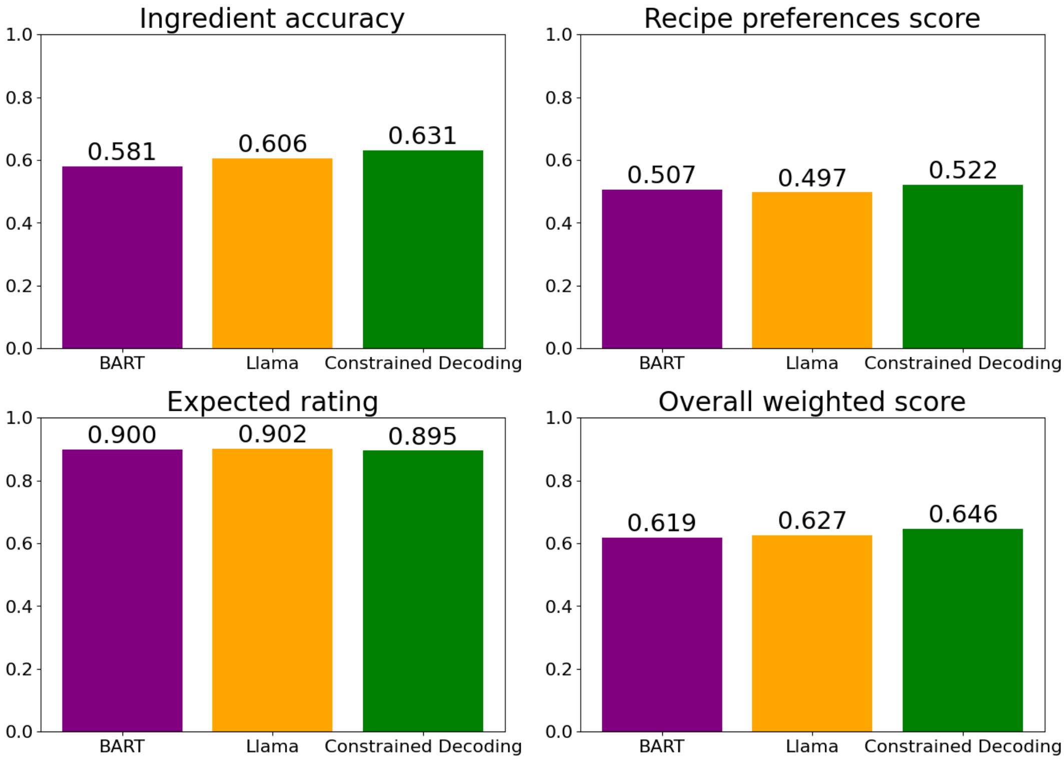

Figure 1: **Performance Metrics for Recipe Generation Models.** Each plot compares the performance of three models: BART (purple), LLaMA3 (yellow), and LLaMA3 with constrained decoding (green). The x-axis represents the model type, while the y-axis indicates the normalized metric score, ranging from 0 to 1. The four metrics displayed are: ingredient accuracy, recipe preferences, expected rating, and overall weighted score. The values above bars are the corresponding metric scores.

LLaMA3, and LLaMA3 with constrained decoding. As shown in Figure 1, the constrained decoding LLaMA3 model achieved the highest ingredient accuracy, 0.631, compared to the baseline BART and LLaMA3 models 0.606 and 0.581 respectively. Next, the top-right quadrant of Figure 1 shows that the constrained decoding LLaMA3 continued to have the highest recipe preferences score at 0.522, but BART's score of 0.507 is slightly higher than LLaMA3's 0.497 for this metric. The bottom-left quadrant of Figure 1 shows that the expected rating for baseline LLaMA3 is highest at 0.902, then BART at 0.900, and then the constrained decoding LLaMA3 model's 0.895. Lastly, the overall weighted average score remains the highest for the constrained decoding LLaMA3 model at 0.646, with LLaMA3 (0.627) and BART (0.619) following closely after.

5.2 Example Recipe

Here, we have provided an example of a recipe generated by the fine-tuned LLaMA3 model with constrained decoding.

```
Input:
Ingredients Required: eggs, noodles,
↪  beef
Preferences:
- Meal Type: Dinner
- Dietary Restrictions: None
- Nutrition: High Protein
- Cuisine: Chinese

Generate a recipe.

Output:

**Beef Noodle Stir-Fry with Egg**

Servings 4
Prep Time 15 minutes
Cook Time 20 minutes
Total Time 35 minutes

**Ingredients**

* 1 cup noodles (preferably rice noodles
↪  or egg noodles)
```

```
* 1 lb beef (sliced into thin strips),
↪  such as ribeye or sirloin
* 2 eggs
* 2 tablespoons vegetable oil
* 2 cloves garlic, minced
* 1 tablespoon soy sauce
* 1 tablespoon oyster sauce (optional)
* 1 teaspoon sesame oil
* Salt and pepper to taste
* Chopped scallions and sesame seeds for
↪  garnish (optional)

**Instructions**

1. Cook the noodles according to the
↪  package instructions. Drain and set
↪  aside.
2. In a small bowl, whisk together soy
↪  sauce, oyster sauce (if using), and
↪  sesame oil. Add the sliced beef and
↪  marinate for at least 5 minutes.
3. Heat 1 tablespoon of vegetable oil in
↪  a large skillet or wok over
↪  medium-high heat. Remove the beef
↪  from the marinade and cook until
↪  browned, about 3-4 minutes. Transfer
↪  the beef to a plate and set aside.
4. In the same skillet, add the
↪  remaining 1 tablespoon of vegetable
↪  oil. Add the minced garlic and
↪  stir-fry for 1 minute, until
↪  fragrant.
5. Push the garlic and oil to one side
↪  of the skillet. Crack in the eggs
↪  and scramble them until cooked
↪  through, about 2-3 minutes.
6. Mix the eggs with the garlic and oil
↪  on the other side of the skillet.
7. Add the cooked noodles to the skillet
↪  and stir-fry for 2-3 minutes, until
↪  the noodles are well coated with the
↪  egg and garlic mixture.
8. Add the marinated beef back to the
↪  skillet and stir-fry for an
↪  additional 1-2 minutes, until the
↪  beef is heated through.
9. Season with salt and pepper to taste.
10. Garnish with chopped scallions and
↪  sesame seeds (if using). Serve hot.

Notes:
* Add a splash of chili oil for spice or
↪  a dash of black vinegar for
↪  tanginess.
* Substitute beef with a vegetarian
↪  protein (e.g. tofu or tempeh) to
↪  modify this recipe for vegetarians.
* For extra crunch, toss in some water
↪  chestnuts or snap peas.
Enjoy your flavorful and hearty meal!
```

6 Discussion and Conclusion

In summary, we found that our recipe generator model of fine-tuned LlaMA3 is the highest performing against our evaluation metrics of ingredient accuracy, recipe preferences, and expected rating. Not only is it the best model overall, but it also performed the best in ingredients accuracy and recipe preferences - both of which were expected effects of constraint decoding. Since the constraint decoding forces the recipes to be in a standardized format that first lists all the ingredients and then details the steps of the recipe, this guides the model to include all of the ingredients that it intends to use in the recipe. This likely explains the increase in ingredient accuracy. We also expected that the recipe preferences would increase with constraint decoding, as constrained decoding directs the recipe to focus on dietary restrictions and cuisine preferences. Although the improvements in ingredient accuracy and recipe preferences came at expense of a decrease in expected rating, we are willing to accept this trade-off as it is our least important metric.

6.1 Limitations and Future Work

We acknowledge that there are limitations in our work due to the scope of our project. Our evaluation results are anchored on the dataset that we use to train our model, as many of our metrics are based on similarity to existing recipes in the dataset. We also would like to improve on how our expected rating metric is calculated. A possible alternative to be implemented in the future would be to directly survey users of the generated recipes to calculate ratings and/or utilize ChatGPT to rate the recipes.

While the current implementation of our constraint decoding significantly improves the alignment of generated recipes with user preferences, it still has several limitations. First, our current lists for cuisines and dietary restrictions were limited to fifty words and may not fully capture the diversity of real-world terms. Changing from word lists to word embeddings could increase the scope of the ingredients we capture. Further, rather than using GPT-generated lists, using frequency analysis on our recipes dataset to generate a list of authentic words could improve the quality of our lists. Finally, all tokens within our list are weighted equally, which may oversimplify the relative importance of specific terms. Introducing weighted prioritization for terms based on their relevance could improve the effectiveness of our constraint decoding. Integrating all of these improvements into our fine-tuning would be promising for future research.

In addition, our compute cluster is not ideal, as Google Colab Pro still has usage limits that we have reached, restricting our ability to train our models to the utmost extent. As a result, we had fewer opportunities to experiment with different architectures and hyperparameters. Rather than being able to tune our hyperparameters to optimize our evaluation like we hoped, we had to restrict ourselves to hyperparameters that would allow us to train our models with our limited resources and time.

To enhance the authenticity of our recipes for non-American cuisines, we would like to also train our models on datasets that exclusively include recipes of a specific non-American cuisine in the future. Incorporating datasets that are in the native language of the corresponding cuisine could even further improve the cultural and culinary authenticity of the generated recipes. To improve our cuisine evaluation metric, we could also seek feedback from native speakers, chefs, or home cooks from the cuisine's culture to validate the authenticity of generated recipes.

In the future, we believe that more work could be done on the novelty aspect of the generated recipes. This is out of scope for this project since our primary focus is evaluating based on user-preference fit and not creativity of the model, but we believe that a large language model that could develop unique recipes, such as fusion cuisine recipes, could lead to new and creative takes on the forefront of recipe generation. By generating recipes that are difficult to find online or are unique from existing recipes, this future model could be useful to users seeking more adventurous culinary experiences.

References

[1] B. Aljbawi, "Health-Aware Food Planner: A Personalized Recipe Generation Approach Based on GPT-2," (dissertation), Wilfrid Laurier University, Waterloo, Ontario, 2020.

[2] H. Lee, K. Shu, P. Achananuparp, P. K. Prasetyo, Y. Liu, E.-P. Lim, and L. R. Varshney, "RecipeGPT: Generative Pre-training Based Cooking Recipe Generation and Evaluation System," *Companion Proceedings of the Web Conference 2020*, 2020. https://doi.org/10.1145/3366424.3383536

[3] R. Kaur, R. Kumar, and M. Gupta, "Deep Neural Network for Food Image Classification and Nutrient Identification: A Systematic Review," *Reviews in Endocrine and Metabolic Disorders*, vol. 24, no. 4, pp. 633–653, 2023. https://doi.org/10.1007/s11154-023-09795-4

[4] P. Krishna and C. Metz, "Can A.I. Write Recipes Better Than Humans? We Put It to the Ultimate Test," *The New York Times*, Nov. 4, 2022. https://www.nytimes.com/2022/11/04/dining/ai-thanksgiving-menu.html

[5] S. Li, "Food.com Recipes and Interactions," *[Data set]*, Kaggle, 2019. https://doi.org/10.34740/KAGGLE/DSV/783630

[6] X. Li, Y. Jiang, and P. Qing, "Estimates of Household Food Waste by Categories and Their Determinants: Evidence from China," *Foods*, vol. 12, no. 4, p. 776, 2023. https://doi.org/10.3390/foods12040776

[7] B. P. Majumder, S. Li, J. Ni, and J. McAuley, "Generating Personalized Recipes from Historical User Preferences," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5975–5981, 2019. https://doi.org/10.18653/v1/d19-1613

[8] DishGen, "Meet Your Personal AI-Powered Kitchen Assistant," *DishGen*, (n.d.). https://www.dishgen.com/

[9] A. Roberts, "Do AI Recipes Work? We Tried 3 AI Recipe Generators to Find Out," *Thrive Market*, Oct. 10, 2024. https://thrivemarket.com/blog/do-ai-recipes-work

[10] J. Saxena, "Go Ahead and Make Your AI Recipe. It Won't Be Good," *Eater*, June 1, 2023. https://www.eater.com/23745417/ai-recipes-are-bad-for-everyone

[11] C.-C. Teng, C. Chih, W.-J. Yang, and C.-H. Chien, "Determinants and Prevention Strategies for Household Food Waste: An Exploratory Study in Taiwan," *Foods*, vol. 10, no. 10, p. 2331, 2021. https://doi.org/10.3390/foods10102331

[12] M. van Erp, C. Reynolds, D. Maynard, et al., "Using Natural Language Processing and Artificial Intelligence to Explore the Nutrition and Sustainability of Recipes and Food," *Frontiers in Artificial Intelligence*, vol. 3, 2021. https://doi.org/10.3389/frai.2020.621577