KERL: *K*nowledge-*E*nhanced Personalized Recipe *R*ecommendation using *L*arge Language Models [*]

Fnu Mohbat and Mohammed J. Zaki
Rensselaer Polytechnic Institute
mohbaf@rpi.edu, zaki@cs.rpi.edu

Abstract

Recent advances in large language models (LLMs) and the abundance of food data have resulted in studies to improve food understanding using LLMs. Despite several recommendation systems utilizing LLMs and Knowledge Graphs (KGs), there has been limited research on integrating food related KGs with LLMs. We introduce KERL, a unified system that leverages food KGs and LLMs to provide personalized food recommendations and generates recipes with associated micro-nutritional information. Given a natural language question, KERL extracts entities, retrieves subgraphs from the KG, which are then fed into the LLM as context to select the recipes that satisfy the constraints. Next, our system generates the cooking steps and nutritional information for each recipe. To evaluate our approach, we also develop a benchmark dataset by curating recipe related questions, combined with constraints and personal preferences. Through extensive experiments, we show that our proposed KG-augmented LLM significantly outperforms existing approaches, offering a complete and coherent solution for food recommendation, recipe generation, and nutritional analysis. Our code and benchmark datasets are publicly available at https://github.com/mohbattharani/KERL.

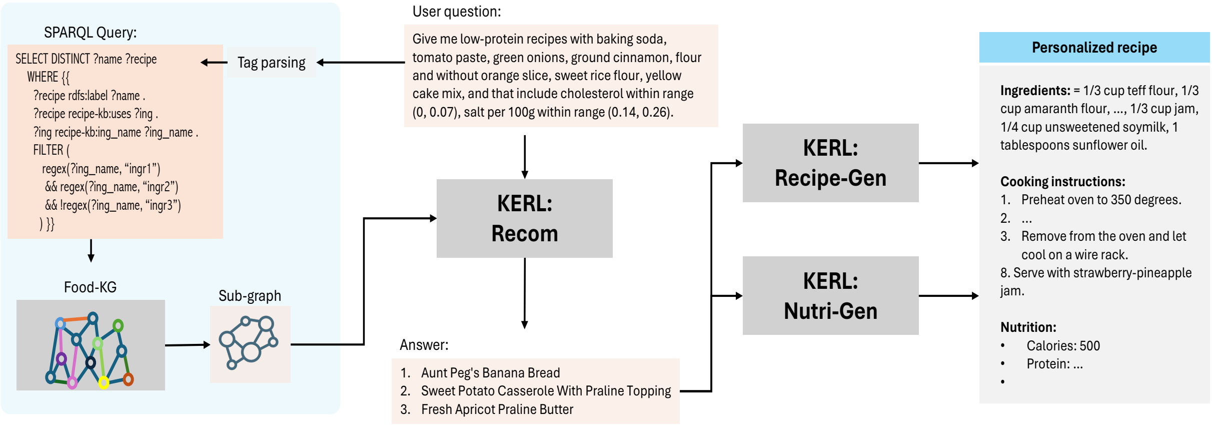
1 Introduction

The importance of food for well-being has created the need to employ machine learning to promote healthy lifestyles through food understanding. Several recipe-sharing websites have created rich resources of food data, attracting researchers to devise food computing for classification, retrieval, recipe generation, and recommendation. Food recommendation is a complex and multifaceted taks given its direct impact on human health. An effective food recommendation system should consider personal preferences, dietary constraints, and health guidelines. In recent years, several ontologies and knowledge graph methods have helped to better organize food data (Dooley et al., 2018; Haussmann et al., 2019; Razzaq et al., 2023). Subsequently, several food recommendation methods have leveraged the KGs for personalized food recommendation (Chen et al., 2021; Shirai et al., 2021; Ling et al., 2022; Li et al., 2023; Kobayashi et al., 2024). Several studies have also utilized LLMs for recipe generation (H. Lee et al., 2020; Yin et al., 2023; Mohbat and Zaki, 2024) and nutrition estimation (Yin et al., 2023; Tanabe and Yanai, 2024, 2025). However, there is a lack of unified food understanding systems that not only recommend personalized recipes but also generate cooking steps and micro-nutrition information for the recommended dishes.

Despite the success of LLMs in multiple domains (Wu et al., 2023; Moor et al., 2023; Chhikara et al., 2024; Mohbat and Zaki, 2024), they are prone to hallucination and outdated information (Xu et al., 2024b). Retrieval-augmented generation (RAG) addresses the issue by utilizing documents or KGs as external knowledge (Mathur et al., 2024; He et al., 2024; Rangel et al., 2024). Question-answering over KGs (KGQA) retrieves the relevant subgraphs from KG, uses reasoning to extract entities as answers (Wang et al., 2021), or uses semantic parsing or LLMs in a zero or few shot setting to transform questions into SQL or SPARQL queries to get answers from the KG (Banerjee et al., 2023; Taffa and Usbeck, 2023; Avila et al., 2024). Despite various attempts to integrate external knowledge with LLMs in several domains, there is a lack of work on food recommendation that combines KGs and LLMs while considering both health constraints and user preferences.

We propose a personalized and unified food recommendation system called KERL that uses the FoodKG (Haussmann et al., 2019) as the knowledge source. The system illustrated in Fig. 1 com-

[*] This paper has been accepted at the ACL 2025.



```
SELECT DISTINCT ?name ?recipe
  WHERE {{
    ?recipe rdfs:label ?name .
    ?recipe recipe-kb:uses ?ing .
    ?ing recipe-kb:ing_name ?ing_name .
    FILTER (
      regex(?ing_name, “ingr1”)
       && regex(?ing_name, “ingr2”)
       && !regex(?ing_name, “ingr3”)
     ) }}
```

Give me low-protein recipes with baking soda, tomato paste, green onions, ground cinnamon, flour and without orange slice, sweet rice flour, yellow cake mix, and that include cholesterol within range (0, 0.07), salt per 100g within range (0.14, 0.26).

1. Aunt Peg's Banana Bread
2. Sweet Potato Casserole With Praline Topping
3. Fresh Apricot Praline Butter

Ingredients: = 1/3 cup teff flour, 1/3 cup amaranth flour, ..., 1/3 cup jam, 1/4 cup unsweetened soymilk, 1 tablespoons sunflower oil.

Cooking instructions:
1. Preheat oven to 350 degrees.
2. ...
3. Remove from the oven and let cool on a wire rack.
8. Serve with strawberry-pineapple jam.

Nutrition:
- Calories: 500
- Protein: ...
-

Figure 1: KERL Overview: Given a natural language question (with constraints), the system parses entities and generates a SPARQL query to retrieve a subgraph from the KG. The question and this subgraph as context, are given as input to the recommendation model (*KERL-Recom*), which generates a list of recipe names that satisfy the constraints. The *KERL-Recipe* and *KERL-Nutri* models then generate cooking steps and micro-nutrients.

prises three modules: a recommendation module (*KERL-Recom*), a recipe generation module (*KERL-Recipe*), and a nutrition generation module (*KERL-Nutri*), which are trained using a low-rank adaption (LoRA) approach (Hu et al., 2022). The *KERL-Recom* module takes a user query, extracts entities, constructs a SPARQL query to retrieve relevant subgraphs from the KG, and inputs these subgraphs along with the query into the LLM, which returns dish names that satisfy the constraints. The *KERL-Recipe* modules generates recipes from the suggested titles, while the *KERL-Nutri* produces detailed micro-nutritional information for recommended dishes. Overall, we make the following contributions.

- We propose KERL, a unified food recommendation system based on a multi-LoRA approach, with a dedicated adapter for each task, while utilizing the same base model, allowing efficient training and inference.
- Our work generates comprehensive nutritional information unlike previous approaches that focus mainly on one aspect, such as calorie count.
- We curated two open benchmark datasets using template questions, nutrient constraints, and personal preferences.
- Through extensive experiments, we show that each module of KERL outperforms the baseline LLMs, showcasing the power of integrating KGs with LLMs.

2 Related Work

Food Recommendation The initial food recommendation systems formulated recommendation as a retrieval task by mapping the recipe components such as title, ingredients, and images into a common embedding space (Salvador et al., 2017; Chen et al., 2018; Wahed et al., 2024; Li and Zaki, 2022). Later, the focus shifted towards the use of food knowledge graphs. For example, (Li and Zaki, 2022; Gao et al., 2022) used graph neural network to learn the user-recipe interactions in KGs, and Chen et al. (2021) proposed knowledge base question answering through information retrieval by mapping questions and possible answers in a common embedding space. However, recent methods employ LLMs for food recommendations. For example, (Kirk et al., 2023) investigated ChatGPT for nutrition questions, and (Geng et al., 2022; Rostami et al., 2024) use LLMs as a language processing engine in the food recommendation system. Despite considerable efforts to leverage KGs and LLMs for developing food recommendation systems, there remains limited research on integrating food KGs to augment LLMs for more personalized food recommendation. Specifically, individual preferences, health considerations, and nutritional constraints within a unified framework have not been extensively explored.

Question Answering Over KGs Question answering over knowledge graphs refers to retrieving knowledge from a KG to answer queries. Initial studies parsed entities from a natural language question and generated SPARQL queries from templates to retrieve the answers (Shirai et al., 2021; Haussmann et al., 2019; Rangel et al., 2024). Later, researchers used embeddings from LSTM or graph neural networks and framed the problem as a retrieval task (Chen et al., 2021; Gao et al., 2022; He et al., 2024). Recent methods explore the integration of KGs to improve LLMs for reasoning (Luo et al., 2023; Sun et al., 2023), chatbots for customer service (Xu et al., 2024a), product recommendations (Eppalapally et al., 2024), and food-related tasks (Qi et al., 2023; Hou and Zhang, 2024; Ma et al., 2024; Zhang et al., 2024). For instance, FoodGPT (Qi et al., 2023) aims to enhance recipe generation, while (Zhang et al., 2024) focuses on recommending foods based on their health effects. Nevertheless, the full potential of KG and LLM integration in food science remains underexplored (Min et al., 2022; Ma et al., 2024), presenting a critical research opportunity.

Recipe Generation One line of research focuses on generating the recipe title from food images or ingredients from the title, and then generating the recipes (Reusch et al., 2021; Chhikara et al., 2024). Several efforts tried to generate recipes directly from inputs such as title, images, and ingredients (Farahani et al., Dec 16, 2023; Yin et al., 2023; Mohbat and Zaki, 2024). RecipeGPT (H. Lee et al., 2020) fine-tunes GPT-2 (Radford et al., 2019) while RecipeMC (Taneja et al., 2024) refines the generated recipes using Monte Carlo Tree Search. RecipeGM (Reusch et al., 2021) and Chef Transformer (Farahani et al., Dec 16, 2023) generate recipes from ingredients, while FIRE (Chhikara et al., 2024) predicts those ingredients from a given image or title. However, more recent methods explore end-to-end fine-tuning of LLMs and multi-modal models (MMMs) for recipe generation. FoodLMM (Yin et al., 2023) fine-tunes LISA (Lai et al., 2024) for classification, ingredient detection, segmentation, and recipe generation, while LLaVA-Chef (Mohbat and Zaki, 2024) investigates better fine-tuning schemes to improve recipe generation. One recent work (Liu et al., 2025) even tried retrieval augmented generation (RAG) for recipe generation. However, all of these methods focus solely on recipe generation.

Nutrition Generation Due to the effectiveness of nutritional intake for personal health, researchers employed MMMs for calorie estimation (Yin et al., 2023; Tanabe and Yanai, 2024; Yao et al., 2024; Tanabe and Yanai, 2025) from food images. Most of these methods aim to estimate the calories from one or more food images utilizing the Nutrition5k (Thames et al., 2021) dataset that contains only 5000 recipes with a total of 125K images. FoodLMM (Yin et al., 2023) leverages LISA (Lai et al., 2024), CalorieLLaVA (Tanabe and Yanai, 2024) fine-tunes LLaVA (Liu et al., 2024) and CaLoRAify (Yao et al., 2024) fine-tunes Llama-2 based visual language model for calorie estimation. Most of the existing work is limited to calorie estimation only, disregarding the estimation of other vital micro-nutrients. This work considers the estimation of several micro-nutrients, including protein, fiber, fat, and cholesterol.

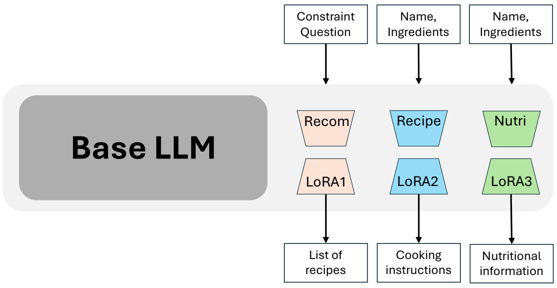


Figure 2: KERL Multi-LoRA Setup: With the same base model, a separate LoRA adapter is trained for each task. During inference, the desired adapter is activated while base model remains the same.

3 KERL: Food Recommendation System

We propose KERL, a personalize food recommendation system that unifies recommendation with cooking steps and nutrition details generation by leveraging multi-LoRA approach as illustrated in Fig. 1. KERL uses the FoodKG (Haussmann et al., 2019) as external knowledge source and an LLM as a generative engine. FoodKG contains 1 million recipes from Recipe1M (Salvador et al., 2017) with ingredients, nutritional information, and tags, organized into 67 million triplets. Let t_j be a tag in FoodKG, and $R(t_j)$ the set of tagged recipes with tag t_j where each recipe X_{recipe} has a title or name X_t, ingredients X_{ing}, cooking steps X_{inst} and nutrients X_{nutri}. Now, let $I(t_j)$ be the set of ingredients in all recipes in $R(t_j)$, then we define $I^+(t_j) \subset I(t_j)$ as the set of ingredients that the user likes to have and $I^-(t_j) \subset I(t_j)$ as the set of ingredients that the user wants to avoid. Let $X_{nutri,i}$ be the i^{th} nutrient, then $\mu(R(t_j), X_{nutri,i})$ and $\sigma(R(t_j), X_{nutri,i})$ denote the mean and standard deviation of the nutrient, respectively. These statistical measures are later used to define nutrient-based preference filters. To incorporate these constraints into personalized recipe recommendations, we employ a modular approach in designing KERL's multi-LoRA architecture, as shown in Fig. 2, where each LoRA adapter is fine-tuned for a distinct sub-task. The *KERL-Recom* adapter is fine-tuned to identify dishes from $R(t_j)$ that satisfy the constraints in the user query as recommended recipes. Subsequently, we also fine-tune the *KERL-Recipe* and *KERL-Nutri* adapters for generation of cooking instructions and micro-nutrients for the recommended recipes. Together, the three modules comprise a comprehensive and integrated food recommendation system.

3.1 KERL-Recom

Given a complex user query containing allowed and disallowed ingredients, and other nutrition-based constraints (see Table 1 for some examples), the main task for the *KERL-Recom* adapter is to recommend relevant recipes leveraging the FoodKG to return a high quality response. The steps involve retrieval of query relevant subgraphs from the KG, fine-tuning the model on user queries, and model inference over the KG for generating the response as recommended recipes.

Subgraph Retrieval: Given a natural language question, the first step is to parse entities such as the tag t_j (e.g., *American*, *Healthy*) and ingredients (e.g., *sugar*, *cheese*). These entities are then used to generate SPARQL queries based on predefined templates, allowing us to retrieve relevant subgraphs from FoodKG. Each subgraph contains the name of the dish, the list of ingredients, and nutritional information. The subgraphs are serialized into a text sequence and given to LLM as a context. An example of a KG subgraph, along with the relevant recipe text is shown in Fig. 4 (See Appendix).

Model Optimization: *KERL-Recom* model is trained to select recipes from the context that meet the constraints in the question. Given that $R(t_j)$ is the set of recipes with the relevant user tag t_j, let $R^+(t_j)$ denote the *positive* subset of recipes that meet all query constraints, and let $R^-(t_j)$ denote the rest of the recipes that make up the *negative* subset. During training, we select a subset of recipes of size at most K, such that we sample at most $K/2$ positive recipes from $R^+(t_j)$ and at most $K/2$ negative recipes from $R^-(t_j)$. This determines the context C_j for the LLM training, along with the full query, with K chosen so that the model can fit within the GPU memory. This approach allows the model to learn to select from a wide distribution of contexts (positive or negative recipes) despite $|R^-(t_j)| \gg |R^+(t_j)|$ (e.g., see the dataset statistics in Table 2). The recipes sampled from $R^+(t_j)$ serve as the ground truth answer Y. The model is trained using low-rank adaptation (Hu et al., 2022) (see details in Sec. 5.1) with standard cross-entropy loss: $L_{CE} = CE(p(Y), p(\tilde{Y}))$, where $p(Y)$ is probability of ground truth recipe tokens as one hot vector and $p(\tilde{Y})$ is the predicted probability of the recipe tokens generated by the model.

Inference over KG: During inference, the entire FoodKG could theoretically be the context for searching relevant recipes. However, in practice, we parse the tag t_j from the query and retrieve $R(t_j)$ as context. The maximum number of tagged recipes could be potentially very large, and the resulting total number of tokens may exceed the LLM's sequence length, which may also lead to GPU memory overflow. Therefore, like in training, we iterate over $R(t_j)$ by providing the LLM with the query and a subset of $R(t_j)$ as context C_j, and combine the responses from multiple calls to the LLM to generate the final answer. This approach allows us to perform inference and evaluate the model on a variable number of recipe subgraphs.

3.2 KERL-Recipe

KERL-Recipe, the recipe generation module enables the recommendation system to generate recipe steps. This module can leverage any recipe generation model, such as LLaVA-Chef (Mohbat and Zaki, 2024) or FoodMMM (Yin et al., 2023). While these models rely on older LLM backbones, recent advances such as LLaMA-3 (Grattafiori et al., 2024) and Phi-3 (Abdin et al., 2024) have significantly outperformed their predecessors. Therefore, we employed Phi-3-mini for recipe generation, specifically generating cooking steps from the dish names X_t, and ingredients X_{ing}, or both. Unlike (Mohbat and Zaki, 2024; Yin et al., 2023), we use LoRA training, which reduces the number of training parameters and decreases the training time. Thus, KERL-Recipe, implemented as a LoRA adapter, integrates seamlessly into the KERL framework, while utilizing Phi-3 Mini as the base model.

Base question: Give me *{tag}* recipes with *{ingredients}* and without *{not_have_ingredients}*	**Base question:** What are the *{tag}* dishes that contain *{ingredients}* but do not contain *{not_have_ingredients}*
Template constraints: have *{nutrition}* no more than *{limit}*, *{nutrition}* within range *{limit}*	**Template constraints:** have *{nutrition}* at least *{limit}*, and *{nutrition}* less than *{limit}*
Personal preferences: tag: low-protein **Likes:** baking soda, tomato paste, green onions, ground cinnamon, flour **Dislikes:** orange slice, rice flour, yellow cake mix **Nutrition constraints:** cholesterol no more than 0.07, salt per 100g (0.14, 0.26)	**Personal preferences: tag:** vegetarian **Likes:** margarine, frozen peas, shredded cheddar cheese, baking soda, vinegar **Dislikes:** cracked wheat, chili pepper, fresh pepper **Nutrition constraints:** fiber at least 4.24, saturated fat less than 6.49
Question: Give me low-protein recipes with baking soda, tomato paste, green onions, ground cinnamon, flour and without orange slice, sweet rice flour, yellow cake mix, and have cholesterol no more than 0.07, salt per 100g within range (0.14, 0.26).	**Question:** What are the top vegetarian recipes containing margarine, frozen peas, shredded cheddar cheese, baking soda, vinegar and excluding cracked wheat, chili pepper, fresh pepper, and meeting the fiber at least 4.24, saturated fat less than 6.49 condition?
Answer: Aunt Pegś Banana Bread, Sweet Potato Casserole With Praline Topping, Fresh Apricot Praline Butter.	**Answer:** B. B. Kingś German Chocolate Cake, Apple Bread, Momś Raisin Rock Cookies

Table 1: Examples of constraints, corresponding questions, and relevant recipe names as ground truth answers. Ingredient preferences specify whether certain ingredients should or should not be included in the recipes. Nutritional constraints are numerical conditions applied to nutrient values, defined by limits such as less than, greater than, or within a specified range.

3.3 KERL-Nutri

KERL-Nutri, the nutrition generation module, is also a LoRA adapter trained to generate micro-nutritional information from the recipe name X_t, the ingredients X_{ing}, and the cooking steps X_{instr} or their combination. The module helps ensure that the recommended recipes follow the nutritional constraints in the user query.

All three modules share the same backbone LLM (namely, Phi-3-mini (Abdin et al., 2024)), with separate LoRA adapters fine-tuned for each task as illustrated in Fig. 2. Training details and hyper-parameters are discussed in Sec. 5.1. This design allows multiple adapters to operate even on a single GPU, enabling practical and efficient inference.

4 Benchmark Generation

One of our contributions is the creation of a large benchmark dataset of realistic constrained user queries for training and evaluation, given the lack of real user data. For *KERL-Recom* and *KERL-Nutri*, we curated base (template) questions using GPT-4 (Achiam et al., 2023), whereas for *KERL-Recipe*, we borrowed template prompts from LLaVA-Chef (Mohbat and Zaki, 2024) (see Sec. 4.4 in Appendix). Based on the task, each base question contains placeholders for inputs which are then replaced with their values.

4.1 Generating Personal Preferences

To personalize the recommendation of recipes, individualized information about the person's likes, dislikes, and other personal choices are important. Our benchmark incorporates both ingredient preferences and nutritional constraints. We combine the base question and constraints to obtain the final constrained question. See Table 1 for examples. Recipes that satisfy all constraints are considered ground truth answers or a positive set of recipes $R^+(t_j)$, and the remaining $R^-(t_j) = R(t_j) - R^+(t_j)$ are considered as a negative set of recipes. This allows us to generate personalized food recommendations that take into account both taste preferences and dietary needs.

Ingredient Preferences: Ingredient preferences consider what a recipe should or should not contain (e.g., *Recipe should contain Spinach and Butter but must not have Nuts*). Given a set of tags, for each tag t_j we create a set of ingredients $I(t_j)$ used by all recipes in $R(t_j)$. To model a person's likes and dislikes of ingredients, we randomly sample two mutually exclusive sets of ingredients $I^+(t_j)$ and $I^-(t_j)$ from $I(t_j)$ such that $I^+(t_j) \cap I^-(t_j) = \emptyset$. One set $I^+(t_j)$ is treated as person's preferred ingredients, while the other set $I^-(t_j)$ is considered as disliked ingredients that one may wish to avoid in the recipes.

Nutritional Constraints: We also generate nutrition-related user preferences by defining constraints on nutrients (e.g.,*recipes with more than 2 grams of protein and less than 500 calories*). Each constraint is defined in the format: <nutrient> <limit> <value> (e.g., salt less than 0.5g). The limit can be one of three filters: 'less than', 'greater than', or 'fall within a defined range'. The <value> represents the threshold for the limit, or a range. To generate nutritional constraints, first we randomly select one of the three filters. Then, we define a threshold for the nutrient x_i^{thresh} by sampling a random number in the range of $\mu(R(t_j), X_{nutri,i}) \pm 2\sigma(R(t_j), X_{nutri,i})$, where μ and σ are the mean and standard deviation. For the range filter, the upper and lower bounds are set as either $(0, x_i^{thresh})$ or $(x_i^{thresh}, max(X_{nutri,i}))$. Finally, all selected nutritional constraints are combined with the base question. This approach enhances the diversity of the questions, incorporating both conditional logic and negations, which are crucial for generating more complex and realistic queries.

Measure	**KGQA Benchmark**		PFoodReq	
	Train set	Test set	Train set	Test set
Number of questions	62320	7790	4613	2305
$R(t_j)$ (min)	7	7	2	2
$R(t_j)$ (max)	4445	4445	2486	2485
$R(t_j)$ (avg)	3167	3163	408.4	377.99
$R^+(t_j)$ (min)	1	1	1	1
$R^+(t_j)$ (max)	1776	954	296	178
$R^+(t_j)$ (avg)	10.67	9.77	2.94	2.84

Table 2: KGQA Benchmark: Total number of questions, and the number of tagged recipes for overall context $R(t_j)$ and ground truth answer $R^+(t_j)$.

4.2 KGQA Benchmark

In the KGQA benchmark, ingredient preferences and nutrition constraints are combined with a user query to create a detailed question. Examples of the base question, constraints, nutritional limits, and the final question are given in Table 1. To generate final queries, we randomly sample a base question from the templates, replace the placeholders with ingredient choices and nutritional constraints. The recipes that meet all the conditions in the final question are considered recommended recipes.

We used FoodKG as our knowledge base, containing over 1 million recipes labeled with 490 unique tags, where each recipe may have multiple tags. Questions were generated based on health-related tags, e.g., dairy-free, low-fat, high-fiber (full list of tags is in Appendix A.1). Our dataset consists of 77,900 question-answer pairs, split into 80% for training, 10% for validation, and 10% for testing. Table 2 shows that the number of recipes for a given tag $R(t_j)$, which is also the possible context size $|C_j|$, ranging from 7 to 4,445, while the recipes in the ground truth answer $R^+(t_j)$ vary from 1 to 954, highlighting the complexity and variety of the questions. Note also, that our KGQA benchmark is over an order of magnitude larger than the pFoodReq dataset (Chen et al., 2021), which has a total of only 6918 questions.

- For <name>, can you calculate the approximate nutritional values for a standard serving?
- Estimated nutritional values for <name>.
- Generate the nutritional values of the dish based on the ingredients: <ingredients>.
- A dish is cooked using <ingredients>, calculate the nutritional values of the dish.
- Generate the nutritional values of the dish based on its step-by-step instructions: <instructions>.
- Based on the cooking instructions provided, calculate the nutritional values of the dish. Instructions: <instructions>.
- For the following dish, estimate the nutritional values. Recipe: <name> <ingredients> <instructions>.

Table 3: Example prompts utilized for training the *KERL-Nutri* model, where placeholders were replaced with the corresponding information.

4.3 Nutrition Generation Benchmark

In the absence of a standardized benchmark for micro-nutrients, we sourced ground-truth micro-nutritional information from Recipe1M (and thus FoodKG) and (Li et al., 2023), resulting in about 500,000 recipe samples for which we were able to gather nutritional information. Using Recipe1M's predefined train-test splits, we use 19,000 recipes for our test set, with the remaining recipes used as the training set for our nutrition generation benchmark. Subsequently, to train LLMs, we curated about 40 template prompts using GPT-4, with examples of some of the prompts given in Table 3. The placeholders <name>, <ingredients>, and <instructions> in the prompts are replaced with their corresponding actual information from the dataset samples. For example, in the prompt "Estimated nutrition for <name>" the placeholder <name> is replaced with the recipe title (name) X_t, which is then input to the LLM to generate the nutritional information. The prompts are intended to generate nutritional information from recipe attributes such as the title X_t, ingredients X_{ing}, and cooking instructions X_{instr}, or their combinations, which allows the model to learn nutritional information from different attributes of the recipes.

4.4 Recipe Generation Benchmark

Recipe generation benchmark utilizes Recipe1M (Salvador et al., 2017), which contains over 1 million recipes, split into train, test and validation sets. The training set consists of 720,639 recipes. For the test set, we use a filtered version of the Recipe1M test set from LLaVA-Chef (Mohbat and Zaki, 2024), referred to as `test50k`, which contains 50,000 recipes. We base our approach on the template prompts used in LLaVA-Chef (Mohbat and Zaki, 2024), which employed GPT-3.5 to generate these prompts. The prompts are designed to generate recipes from a given title (X_t), a list of ingredients (X_{ing}), or both. The example prompts are provided in Table 4.

- Generate a comprehensive recipe for crafting <name>.
- Detail the method for cooking a delightful <name>.
- Construct a detailed cooking procedure for <name>.
- Generate a recipe using <ingredients>.
- Given <ingredients>, give the detailed recipe.
- Compose a recipe for making a dish using the ingredients: <ingredients>.
- Generate a recipe for crafting <name> using <ingredients>.
- Outline the process of making a delicious <name> using <ingredients>
- Given <ingredients>, suggest me recipe of <name>

Table 4: Example prompts utilized training *KERL-Recipe* model, where placeholders were replaced with the corresponding information.

5 Experimental Results

For baseline comparison, we select several open source LLMs, as detailed Appendix B. We report the performance of *KERL-Recom* on standard retrieval metrics such as precision, recall, and F1, and *KERL-Recipe* on various text generation and summarization metrics including BLEU (Papineni et al., 2002), Rouge (Lin, 2004), METEOR (Elliott and Keller, 2013) and CIDer (Vedantam et al., 2015). For *KERL-Nutri*, we parse micro-nutrients from the generated response and compute the mean average error (MAE) with ground truth. The metric definitions are provided in Appendix C. Our code and benchmark datasets can be found at `https://github.com/mohbattharani/KERL`.

5.1 Experimental Setup

We leverage Phi-3-mini for its performance and compact size and fine-tuned one LoRA (Hu et al., 2022) adapter per task. For each task, we used the same LoRA configuration with $r = 64$ and $\alpha = 16$ and $dropout = 0.5$, where r is the dimensionality of the low rank and α is the scaling factor. We trained a separate LoRA adapter for each task, the overall model is shown in Figure 2. During inference, the same model with multiple adapters allows us to deploy once and activate the task-specific adapter as needed. All experiments were performed on four NVIDIA RTX A6000 GPUs. Each LoRA adapter is trained for two epochs on task related dataset. The training hyper parameters were kept the same for all models, with a starting learning rate of $lr = 2\times10^{-5}$ and a cosine learning rate scheduler. During validation, hyperparameters were also fixed for all the models. Specifically, we used `temperature = 0.2`, `num beams =1` and maximum new tokens to 1024.

Model	mAP	P	R	F1
internLM2 (Cai et al., 2024)	0.06	0.024	0.055	0.034
Mistral (Jiang et al., 2023)	0.214	0.536	0.558	0.547
Phi-2 (Javaheripi et al., 2023)	0.271	0.084	0.378	0.137
Llama-2 (Touvron et al., 2023)	0.557	0.825	0.627	0.713
Llama-3.1 (Grattafiori et al., 2024)	0.146	0.28	0.406	0.332
Phi-3-mini-4K (Abdin et al., 2024)	0.047	0.192	0.044	0.071
Phi-3-mini-128K (Abdin et al., 2024)	0.275	0.778	0.278	0.41
KERL-Recom	**0.96**	**0.978**	**0.969**	**0.973**

Table 5: KGQA Benchmark Test Set: *KERL-Recom* versus pre-trained LLMs.

5.2 *KERL-Recom* Evaluation

Comparison with Open Source LLMs: Table 5 presents the results of recent state-of-the-art LLMs for recipe recommendation. Despite claims of superiority by internLM2 (Cai et al., 2024) and Llama-3.1 (Grattafiori et al., 2024) on various benchmarks, both failed to understand the complex constraints in our KGQA benchmark questions. The capability of handing larger sequence length by Phi-3-mini-128K (Abdin et al., 2024) helps it perform better than Phi-3-mini-4K. Therefore, we selected Phi-3-mini-128K as the base LLM for KERL due to its compact size (3.8B parameters) and competitive performance. Our fine-tuned LoRA *KERL-Recom* model significantly outperforms the other models, achieving a 56-point improvement over Phi-3-mini-128K and 26-point improvement over the larger Llama-2-7B (Touvron et al., 2023) in F1 score.

Impact of Recipe Types: To evaluate the generalization across various types of recipes, we compare

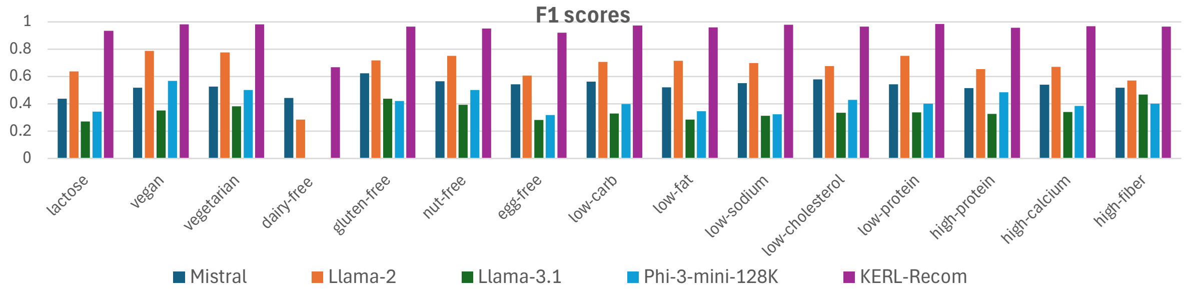


Figure 3: F1 scores of different models across various recipe types. Our model, KERL-Recom, consistently outperforms others with a significant margin in all categories.

Tag	mAP	P	R	F1
lactose	0.898	0.955	0.916	0.935
vegan	0.964	0.988	0.975	0.981
vegetarian	0.966	0.987	0.976	0.981
dairy-free	0.667	0.667	0.667	0.667
gluten-free	0.922	0.992	0.779	0.873
nut-free	1.0	0.909	1.0	0.952
egg-free	0.939	0.982	0.951	0.966
low-carb	0.952	0.983	0.965	0.974
low-fat	0.964	0.956	0.966	0.961
low-protein	0.981	0.988	0.981	0.984
low-sodium	0.982	0.978	0.982	0.98
low-cholesterol	0.924	0.98	0.951	0.965
high-protein	0.992	0.944	0.967	0.956
high-calcium	0.937	0.981	0.953	0.967
high-fiber	0.938	1.0	0.933	0.966

Table 6: *KERL-Recom*: per tag results on KGQA test set

F1 score for the baseline models and *KERL-Recom* in Fig. 3 (and Table 13 in Appendix); our model consistently outperforms all others. For *dairy-free* recipes, only Llama-2 and our model could recommend the correct recipes. Furthermore, the similarly high scores for *KERL-Recom* for most recipe types, except for *dairy-free* and *gluten-free*, as shown in Table 6 indicates that it generalizes well across different types of dishes. Relatively lower accuracy and F1 scores for *dairy-free* recipes is due to the dearth of training samples of this tag (see Table 11 in Appendix).

Comparison on pFoodReq Benchmark: The PFoodReq approach (Chen et al., 2021) can also generate recommendations for constrained queries. However, it does it via an embedding-based approach that computes the similarity between the user query embedding and KG subgraph embeddings. Table 7 shows how our KG-enhanced LLM approach performs on the pFoodReq benchmark dataset. We see that *KERN-Recom* outperforms pFoodRec by 21.7 points on F1; it also outperforms Llama-2-7B by 56.5 points. Our method by utilizing the power of generative models generalizes better and outperforms the classical embedding based methods.

Model	mAP	P	R	F1
P-MatchNN	0.455	-	0.451	0.412
pFoodReq	0.627	-	0.618	0.637
Llama-2	0.322	0.204	0.498	0.289
KERL-Recom	**0.769**	**0.825**	**0.885**	**0.854**

Table 7: PFoodReq Results: Our model compared to baseline shows better scores.

5.3 *KERL-Recipe* Evaluation

Given recipe titles X_t, recipe ingredients X_{ing} and recipe images X_i, or subsets thereof, we now evaluate how well models can generate the actual recipe cooking steps X_{inst}. We compare pretrained LLMs and their fine-tuned counterparts for recipe generation in Table 8. LLaVA-Chef (Mohbat and Zaki, 2024), based on LLaVA, is a state-of-the-art model for this task, and it shows better scores than the pretrained LLaVA and LLaMA baselines. However, we can observe that *KERL-Recipe*, not only outperforms its base Phi-3 model, it has the best overall performance along various recipe quality metrics. It outperforms LLaVA-Chef, which involves entire model training, whereas *KERL-Recipe* is only LoRA fine-tuned. LLaVA-Chef improves almost 7 points on BLEU-1 over its base LLaVA model, whereas *KERL-Recipe* improves about 20 points over Phi-3. Both LLaVA-Chef and *KERL-Recipe* perform better when provided with ingredients X_{ing}, compared to only using the recipe title (X_t), suggesting that the ingredients are important in recipe generation. Note that LLaVA-Chef has the ability to process images, which enables it to generate recipes from food images, whereas *KERL-Recipe* is a text-only model. Overall, *KERL-Recipe* not only outperforms in most metrics, it requires training fewer parameters (LoRA adapter), yet improves over its base model with significant margin.

Model	Inputs	BLEU-1	BLEU-2	BLEU-3	BLEU-4	SacreBLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	CIDEr	Perplexity ↓
LLaMA (Touvron et al., 2023)	X_t + X_{ing}	0.252	0.129	0.072	0.043	0.053	0.156	0.293	0.077	0.156	0.031	2.86
LLaVA (Liu et al., 2024)	X_i + X_t + X_{ing}	0.290	0.155	0.087	0.051	0.06	0.20	0.366	0.105	0.184	0.041	12.39
LLaVA-Chef	X_t	0.283	0.149	0.081	0.047	0.116	0.142	0.37	0.108	0.193	0.094	2.08
LLaVA-Chef	X_i + X_{ing}	0.337	0.197	0.12	0.077	0.156	0.177	0.45	0.156	0.232	0.203	2.43
LLaVA-Chef	X_i + X_t + X_{ing}	0.366	0.218	0.137	0.09	0.170	0.189	**0.473**	0.17	0.240	0.242	17.90
Phi-3	X_t	0.178	0.089	0.047	0.025	0.029	0.187	0.268	0.069	0.134	0.003	11.51
Phi-3	X_{ing}	0.202	0.108	0.06	0.034	0.039	0.207	0.298	0.087	0.149	0.003	11.65
Phi-3	X_t + X_{ing}	0.209	0.114	0.064	0.038	**0.216**	0.042	0.31	0.095	0.155	0.	11.99
KERL-Recipe	X_t	0.292	0.155	0.089	0.053	0.072	0.132	0.317	0.09	0.171	0.10	7.60
KERL-Recipe	X_{ing}	0.392	0.249	0.170	0.12	0.15	0.188	0.441	0.179	0.239	0.323	8.29
KERL-Recipe	X_t + X_{ing}	**0.405**	**0.257**	**0.175**	**0.123**	0.154	**0.195**	0.454	**0.183**	**0.241**	**0.347**	7.68

Table 8: Performance on Recipe Generation

Model	Inputs	Calories	Fat Calories	Protein	Sugar	Fiber	Carbohydrates	Sodium	Cholesterol	Saturated Fat	Total Fat
Dataset Mean		426.39± 626.7	189.79±332.39	15.83±21.26	18.56± 52.27	3.55± 5.56	43.64± 80.57	0.66± 2.03	0.08± 0.14	8.15± 16.49	21.14±36.93
LLaVA-Chef	X_t	306.54	172.16	11.8	17.44	3.85	33.31	29.02	3.13	10.46	23.4
LLaVA-Chef	X_{ing}	323.75	160.45	15.47	20.96	7.48	38.39	160.72	16.82	19.81	37.61
LLaVA-Chef	$X_{instruct}$	319.42	161.77	15.39	22.31	7.86	38.87	111.78	14.63	19.76	36.94
LLaVA-Chef	X_t + X_{ing} + X_{inst}	323.6	161.5	12.68	20.6	5.43	39.29	192.55	21.51	19.78	40.32
KERL-Nutri	X_t	258.28	134.81	8.97	14.22	2.32	29.05	0.5	0.05	6.12	14.98
KERL-Nutri	X_{ing}	226.65	104.98	7.44	12.28	1.88	25.38	0.38	0.04	4.56	11.67
KERL-Nutri	$X_{instruct}$	245.54	124.7	8.58	13.58	2.19	27.68	0.46	0.04	5.45	13.86
KERL-Nutri	X_t + X_{ing} + X_{inst}	**221.38**	**103.03**	**7.29**	**11.92**	**1.84**	**24.69**	**0.37**	**0.04**	**4.48**	**11.45**

Table 9: Performance on Nutrient Generation: Mean absolute error per micro-nutrient.

5.4 *KERL-Nutri* Evaluation

Nutrition generation module (*KERL-Nutri*) is based on LoRA fine-tuning the base Phi-3 model. For comparison, we also fine-tune LLaVA-chef (the full model) on the nutrition generation benchmark. Our model outperforms others, as evident in Table 9 where the first row shows the mean values of the micro-nutrients in the test set. LLaVA-Chef estimates nutrition slightly better when only title X_t is given compared to using ingredients X_{ing} or instructions $X_{instruct}$. However for *KERL-Nutri*, ingredients play a crucial role in nutrition estimation, as they contain the actual nutrients. Generating nutrients from only instruction has slightly higher MAE, as instructions may not explicitly mention all ingredients. Overall, *KERL-Nutri* achieves lower errors when provided with the complete recipe, including the title, ingredients, and cooking instructions.

6 Conclusion

We present KERL, a food recommendation system that combines the power of KGs with LLMs in a question answering framework. We also create a large-scale QA benchmark dataset using FoodKG. After evaluation of several open source LLMs we selected Phi-3-mini as the base LLM, training it to understand the subgraphs from FoodKG to help answer complex constrained questions regarding personalized food recommendations. Using a multi-LoRA approach, we also fine-tune adapters to generate cooking steps and nutritional information for the recipes, offering a seamless solution for meal planning and cooking. Our evaluation shows that KERL outperforms baseline models for all three tasks, with more relevant recipes, better quality cooking steps, and more accurate nutrient values. In the future, we plan to leverage Chain-of-Thought reasoning along with RAG to further improve the performance while incorporating ingredient substitution, person's health information, and cultural preferences.

7 Limitations

- *KERL-Recom* relies on the recipe subgraphs retrieved from FoodKG (Haussmann et al., 2019). Therefore, the system will not recommend any recipe if none of the recipe in KG meet all the constraints. The system may also fail if incorrect context information is provided, hence the results should not be used without proper safeguards.

- *KERL-Recom* do not directly establish the relationship between the person's health conditions and the corresponding dietary restrictions. For example, it can recommend sugar-free recipes, but it may not accurately recommend the correct recipes for a diabetic person. This capability is left for future research.

- *KERL-Nutri* generates the micro-nutritional information for most recipes, but it may not accurately generate micro-nutritional details for extreme cases, such as for recipes with high or very low calories.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report.

Caio Viktor S Avila, Vânia MP Vidal, Wellington Franco, and Marco A Casanova. 2024. Experiments with text-to-sparql based on chatgpt. In *The 18th International Conference on Semantic Computing*, pages 277–284. IEEE.

Debayan Banerjee, Sushil Awale, Ricardo Usbeck, and Chris Biemann. 2023. Dblp-quad: A question answering dataset over the dblp scholarly knowledge graph. *13th International Workshop on Bibliometric-enhanced Information Retrieval*.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

Jing-Jing Chen, Chong-Wah Ngo, Fu-Li Feng, and Tat-Seng Chua. 2018. Deep understanding of cooking procedure for cross-modal recipe retrieval. In *Proceedings of the 26th ACM International Conference on Multimedia*.

Yu Chen, Ananya Subburathinam, Ching-Hua Chen, and Mohammed J Zaki. 2021. Personalized food recommendation as constrained question answering over a large-scale food knowledge graph. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 544–552.

Prateek Chhikara, Dhiraj Chaurasia, Yifan Jiang, Omkar Masur, and Filip Ilievski. 2024. Fire: Food image to recipe generation. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 8184–8194.

Damion M Dooley, Emma J Griffiths, Gurinder S Gosal, Pier L Buttigieg, Robert Hoehndorf, Matthew C Lange, Lynn M Schriml, Fiona SL Brinkman, and William WL Hsiao. 2018. Foodon: a harmonized food ontology to increase global food traceability, quality control and data integration. *NPJ Science of Food*, 2:23.

Desmond Elliott and Frank Keller. 2013. Image description using visual dependency representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Swetha Eppalapally, Daksh Dangi, Chaithra Bhat, Ankita Gupta, Ruiyi Zhang, Shubham Agarwal, Karishma Bagga, Seunghyun Yoon, Nedim Lipka, Ryan A Rossi, et al. 2024. Kapqa: Knowledge-augmented product question-answering. *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*.

Mehrdad Farahani, Kartik Godawat, Haswanth Aekula, Deepak Pandian, and Nicholas Broad. Dec 16, 2023. Chef transformer. *https://huggingface.co/flax-community/t5-recipe-generation*.

Xiaoyan Gao, Fuli Feng, Heyan Huang, Xian-Ling Mao, Tian Lan, and Zewen Chi. 2022. Food recommendation with graph convolutional network. *Information Sciences*, 584:170–183.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Helena H. Lee, Ke Shu, Palakorn Achananuparp, Philips Kokoh Prasetyo, Yue Liu, Ee-Peng Lim, and Lav R Varshney. 2020. Recipegpt: Generative pre-training based cooking recipe generation and evaluation system. In *Companion Proceedings of the Web Conference*.

Steven Haussmann, Oshani Seneviratne, Yu Chen, Yarden Ne'eman, James Codella, Ching-Hua Chen, Deborah L McGuinness, and Mohammed J Zaki. 2019. Foodkg: a semantics-driven knowledge graph for food recommendation. In *The Semantic Web–ISWC: 18th International Semantic Web Conference, Auckland, New Zealand*, pages 146–162. Springer.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.

Yu Hou and Rui Zhang. 2024. Enhancing dietary supplement question answer via retrieval-augmented generation (rag) with llm. *medRxiv*, pages 2024–09.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1:3.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Daniel Kirk, Elise van Eijnatten, and Guido Camps. 2023. Comparison of answers between chatgpt and human dieticians to common nutrition questions. *Journal of Nutrition and Metabolism*, 2023:5548684.

Akio Kobayashi, Shotaro Mori, Akira Hashimoto, Tetsuo Katsuragi, and Takahiro Kawamura. 2024. Functional food knowledge graph-based recipe recommendation system focused on lifestyle-related diseases. In *18th International Conference on Semantic Computing*, pages 261–268. IEEE.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. 2024. Lisa: Reasoning segmentation via large language model. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 9579–9589.

Diya Li and Mohammed J Zaki. 2022. Food knowledge representation learning with adversarial substitution. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*.

Diya Li, Mohammed J Zaki, and Ching-hua Chen. 2023. Health-guided recipe recommendation over knowledge graphs. *Journal of Web Semantics*, 75:100743.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 605–612.

Yabo Ling, Jian-Yun Nie, Daiva Nielsen, Bärbel Knäuper, Nathan Yang, and Laurette Dubé. 2022. Following good examples-health goal-oriented food recommendation based on behavior data. In *Proceedings of the ACM Web Conference 2022*, pages 3745–3754.

Guoshan Liu, Hailong Yin, Bin Zhu, Jingjing Chen, Chong-Wah Ngo, and Yu-Gang Jiang. 2025. Retrieval augmented recipe generation. In *2025 Winter Conference on Applications of Computer Vision*, pages 2453–2463. IEEE.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*.

Peihua Ma, Shawn Tsai, Yiyang He, Xiaoxue Jia, Dongyang Zhen, Ning Yu, Qin Wang, Jaspreet KC Ahuja, and Cheng-I Wei. 2024. Large language models in food science: Innovations, applications, and future. *Trends in Food Science & Technology*, page 104488.

Puneet Mathur, Zhe Liu, Ke Li, Yingyi Ma, Gil Karen, Zeeshan Ahmed, Dinesh Manocha, and Xuedong Zhang. 2024. Doc-rag: Asr language model personalization with domain-distributed co-occurrence retrieval augmentation. In *Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 5132–5139.

Weiqing Min, Chunlin Liu, Leyi Xu, and Shuqiang Jiang. 2022. Applications of knowledge graphs for food science and industry. *Patterns*, 3.

Fnu Mohbat and Mohammed J Zaki. 2024. Llava-chef: A multi-modal generative model for food recipes. In *ACM International Conference on Information and Knowledge Management*.

Michael Moor, Qian Huang, Shirley Wu, Michihiro Yasunaga, Yash Dalmia, Jure Leskovec, Cyril Zakka, Eduardo Pontes Reis, and Pranav Rajpurkar. 2023. Med-flamingo: a multimodal medical few-shot learner. In *Machine Learning for Health (ML4H)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Zhixiao Qi, Yijiong Yu, Meiqi Tu, Junyi Tan, and Yongfeng Huang. 2023. Foodgpt: A large language model in food testing domain with incremental pre-training and knowledge graph prompt. *arXiv preprint arXiv:2308.10173*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Julio C Rangel, Tarcisio Mendes de Farias, Ana Claudia Sima, and Norio Kobayashi. 2024. Sparql generation: an analysis on fine-tuning openllama for question answering over a life science knowledge graph. *arXiv preprint arXiv:2402.04627*.

Muhammad Saad Razzaq, Fahad Maqbool, Muhammad Ilyas, and Hajira Jabeen. 2023. Evorecipes: A generative approach for evolving context-aware recipes. *IEEE Access*.

Anja Reusch, Alexander Weber, Maik Thiele, and Wolfgang Lehner. 2021. Recipegm: A hierarchical recipe generation model. In *The 37th International Conference on Data Engineering Workshops*, pages 24–29. IEEE.

Ali Rostami, Ramesh Jain, and Amir M Rahmani. 2024. Food recommendation as language processing (f-rlp): A personalized and contextual paradigm. *arXiv preprint arXiv:2402.07477*.

Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. 2017. Learning cross-modal embeddings for cooking recipes and food images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE.

Sola S Shirai, Oshani Seneviratne, Ching-Hua Chen, Daniel M Gruen, and Deborah L McGuinness. 2021. Healthy food recommendation and explanation generation using a semantically-enabled framework? In *International Semantic Web Conference: Posters, Demos, and Industry Tracks*. CEUR-WS.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. *arXiv preprint arXiv:2307.07697*.

Tilahun Abedissa Taffa and Ricardo Usbeck. 2023. Leveraging llms in scholarly knowledge graph question answering. In *Scholarly QALD Challenge at The 22nd International Semantic Web Conference*.

Hikaru Tanabe and Keiji Yanai. 2024. Caloriellava: Image-based calorie estimation with multimodal large language models. In *Proceedings of the Proceedings of ICPR Workshop on Multimedia Assisted Dietary Management, Kolkata, India*, volume 1.

Hikaru Tanabe and Keiji Yanai. 2025. Calorievol: Integrating volumetric context into multimodal large language models for image-based calorie estimation. In *International Conference on Multimedia Modeling*, pages 353–365. Springer.

Karan Taneja, Richard Segal, and Richard Goodwin. 2024. Monte carlo tree search for recipe generation using gpt-2. *The 14th International Conference on Computational Creativity*.

Quin Thames, Arjun Karpur, Wade Norris, Fangting Xia, Liviu Panait, Tobias Weyand, and Jack Sim. 2021. Nutrition5k: Towards automatic nutritional understanding of generic food. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 8903–8911.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 4566–4575. IEEE.

Muntasir Wahed, Xiaona Zhou, Tianjiao Yu, and Ismini Lourentzou. 2024. Fine-grained alignment for cross-modal recipe retrieval. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 5584–5593. IEEE.

Haiwen Wang, Le Zhou, Weinan Zhang, and Xinbing Wang. 2021. Literatureqa: A qestion answering corpus with graph knowledge on academic literature. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4623–4632.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. 2024a. Retrieval-augmented generation with knowledge graphs for customer service question answering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2905–2909.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024b. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Dongyu Yao, Keling Yao, Junhong Zhou, and Yinghao Zhang. 2024. Caloraify: Calorie estimation with visual-text pairing and lora-driven visual language models. *arXiv preprint arXiv:2412.09936*.

Yuehao Yin, Huiyan Qi, Bin Zhu, Jingjing Chen, Yu-Gang Jiang, and Chong-Wah Ngo. 2023. Foodlmm: A versatile food assistant using large multi-modal model. *arXiv preprint arXiv:2312.14991*.

Zheyuan Zhang, Zehong Wang, Tianyi Ma, Varun Sameer Taneja, Sofia Nelson, Nhi Ha Lan Le, Keerthiram Murugesan, Mingxuan Ju, Nitesh V Chawla, Chuxu Zhang, et al. 2024. Mopi-hfrs: A multi-objective personalized health-aware food recommendation system with llm-enhanced interpretation. *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

A Dataset Details

A.1 Recipe tags

Recipes in FoodKG are tagged with one or more tags, making a total of 490 unique tags. We selected recipes associated with health-related tags, listed in Table 10, for the generation of KGQA benchmark to ensure that the question-answer pairs inherently focus on health constraints. Note that the recipes tagged with these tags were also tagged with 454 other tags, indicating that the dataset covers a wide variety of recipe types. An example of a KG subraph correponding to a recipe is shown in Fig. 4.

lactose	vegan	vegetarian
dairy-free	gluten-free	nut-free
egg-free	low-carb	low-fat
low-sodium	low-cholesterol	low-protein
high-protein	high-calcium	high-fiber

Table 10: Tags representing various dietary preferences and nutritional constraints.

A.2 KGQA benchmark details

The recipes in FoodKG, the knowledge base used for KGQA benchmark, are tagged with one or more tags. Table 11 shows the number of tagged recipes for each tag used for dataset generation and the associated number of questions in the train and test splits. The limited number of dairy-free tagged recipes (only 7) led to fewer corresponding test questions (only 3). As a result, the evaluated models also show the lowest F1 score for this tag, as shown in Figure 3.

Tag	Tagged recipes	Train set	Test set
lactose	366	874	112
vegan	968	2287	314
vegetarian	3392	8142	979
dairy-free	7	18	3
gluten-free	565	1328	187
nut-free	45	107	13
egg-free	440	1078	124
low-carb	4239	10248	1244
low-fat	2202	5296	639
low-sodium	4445	10561	1407
low-cholesterol	3710	8990	1059
low-protein	3320	7944	1032
high-protein	690	1642	219
high-calcium	540	1318	162
high-fiber	33	76	8

Table 11: KGQA benchmark: Number of tagged recipes for each tag and questions for each tag.

B Foundational Models

Here we provide a list of baseline LLMs we compare with in our empirical evaluation.

internLM2 (Cai et al., 2024) is the second generation internLM model, trained to capture long-term dependencies. It outperforms on 30 benchmarks in long context modeling and open-ended subjective evaluations.

Mistral (Jiang et al., 2023) is engineered for superior performance and efficiency. Its 7B model can outperforms LLaMA-2 13B model.

LLama-2 (Touvron et al., 2023) is a collection of foundation language models ranging from 7B to 70B. Due to the popularity of the llama series, we select Llama-2-7B model in our study.

Llama-3.1 (Grattafiori et al., 2024) is a set of large scale very powerful open source LLM that improves upon Llama-2, and is comparable to the flagship models like GPT-4 and Claude 3.5 Sonnet. Therefore, it became an obvious choice for our study.

Phi-2 (Javaheripi et al., 2023) is a 2.7B parameter LLMs designed for efficient and high-performing natural language processing tasks. It has demonstrated better performance than the LLaMA-2 (13B) and Mistral (7B) models on a range of benchmark tasks, showcasing its effectiveness in various NLP domains.

Phi-3 (Abdin et al., 2024) has improved models in the Phi series; even its mini version with 3.8B parameters outperforms several 7B and 13B models. We used Phi-3-mini-4k and Phi-3-mini-128K in our study for their performance despite their smaller size.

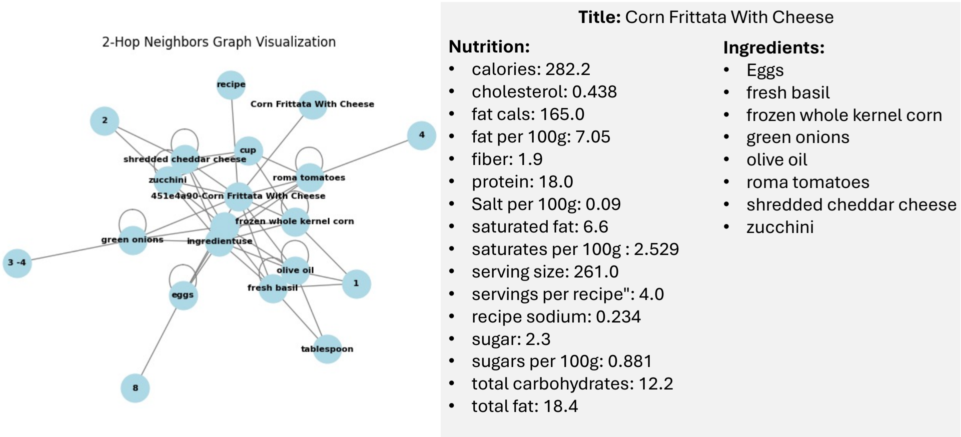
C Metrics

C.1 Metrics for Recommendation Evaluation

We use standard retrieval metrics and provide their formal definitions considering order agnostic evaluation of all the models. Let Y be a list of recipes as ground truth answer and $\tilde{Y}$ a list of recipes recommended by the model. Then, we define true positive (TP), false positive (FP) and false negative (FN) as follows:

$$TP = Y \cap \tilde{Y}$$
$$FP = \tilde{Y} - Y$$
$$FN = Y - \tilde{Y}$$

Then precision (P), recall (R), and F1 scores are computed as follows:



Title: Corn Frittata With Cheese

Nutrition:
- calories: 282.2
- cholesterol: 0.438
- fat cals: 165.0
- fat per 100g: 7.05
- fiber: 1.9
- protein: 18.0
- Salt per 100g: 0.09
- saturated fat: 6.6
- saturates per 100g : 2.529
- serving size: 261.0
- servings per recipe": 4.0
- recipe sodium: 0.234
- sugar: 2.3
- sugars per 100g: 0.881
- total carbohydrates: 12.2
- total fat: 18.4

Ingredients:
- Eggs
- fresh basil
- frozen whole kernel corn
- green onions
- olive oil
- roma tomatoes
- shredded cheddar cheese
- zucchini

Figure 4: FoodKG Recipe sample: left panel shows a 2-hop KG subgraph for the recipe node shown on the right.

$$P = \frac{|TP|}{|TP| + |FP|}$$
$$R = \frac{|TP|}{|TP| + |FN|}$$
$$F1 = \frac{2PR}{P + R}$$

We compute precision at rank r for all relevant recipes M and average them to get average precision (AP). Then, we calculate mean average precision (mAP) by taking the average of AP across all N samples, formally defined as follows:

$$AP = \frac{1}{|M|} \sum_{r \in M} P(r)$$
$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

C.2 Metrics for Recipe Generation

Here we provide formal definitions of the different metrics used in our evaluation of generated recipes.

BLEU score: Bilingual Evaluation Understudy (BLEU) score (Papineni et al., 2002), initially proposed for machine translation evaluation, is a metric that quantifies the similarity between a generated text sequence and a reference text sequence. Let Y_{pred} be the predicted text sequence of length n_p, Y_{label} be the ground truth text sequence of length n_l, and N the number of n-grams, then the BLEU score is defined as:

$$BLEU = BP \exp \sum_{n=1}^{N} w_n \cdot \log(prec)$$

$$prec = \frac{\sum_{p \in Y_{pred}} \sum_{n\text{-}gram \in p} Count_{clip}(n\text{-}gram)}{\sum_{p' \in Y_{pred}} \sum_{n\text{-}gram' \in p'} Count(n\text{-}gram')}$$

$$BP = \begin{cases} 1, & n_p > n_l \\ e^{1 - n_p/n_l} & n_p \leq n_l \end{cases}$$

Where, $prec$ is n-gram precision, w_n is the weight for each precision score and BP is brevity penalty that penalizes too short sequences. For the BLEU-N score, the weight of the precision score is $w_n = \frac{1}{N}$. BLEU cannot be directly compared between research papers (Post, 2018) as it is a parameterized metric and the parameters are often not reported. Therefore, we adhere to the standard implementation of BLEU-N (https://github.com/salaniz/pycocoevalcap). Furthermore, SacreBLEU (Post, 2018) (https://github.com/mjpost/sacreBLEU) is a reproducible and shareable implementation of the BLEU score.

Rouge score: Rouge (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) was designed to evaluate the text summarization system. Rouge-N computes N-gram recall between predicted text (summary) and ground truth, and is defined as:

$$Rouge\text{-}N = \frac{\sum_{s \in Y_{label}} \sum_{n\text{-}gram \in s} Count_{match}(n\text{-}gram)}{\sum_{s \in Y_{label}} \sum_{n\text{-}gram \in s} Count(n\text{-}gram)}$$

Where $Count_{match}$ is maximum number of n-grams co-occurring in predicted text Y_{pred} and reference ground truth Y_{label}. Rouge-L (Lin and Och, 2004) operates on the basis of the longest common subsequence between generated text and ground truth references. It measures the extent to which the generated text captures the longest in-sequence co-occurrences of words in the references.

CIDEr: Consensus-based Image Description Evaluation (CIDEr) (Vedantam et al., 2015) was introduced as a metric to quantify the quality of generated image captions or descriptions. It operates by measuring the degree of consensus between a generated caption and a set of human-authored reference captions. The mathematical formulation of CIDEr is as follows:

$$CIDEr(Y_{pred}, Y_{label}) = \\ \sum_{n=1}^{N} w_n \frac{1}{m} \sum_{j} \frac{g^n(Y_{pred}).g^n(Y^j_{label})}{||g^n(Y_{pred})||||g^n(Y^j_{label})||}$$

where Y^j_{label} is j^{th} ground truth sentence, $g^n(.)$ is a vector representing Term Frequency Inverse Document Frequency (TF-IDF) weighting for each n-gram.

METEOR: METEOR (Metric for Evaluation of Translation with Explicit Ordering) (Elliott and Keller, 2013) is a machine translation evaluation metric that leverages unigram matching between machine-generated translations (hypotheses) and human-produced references (ground truth). It incorporates both precision (*P*) and recall (*R*) of unigrams, as well as other features such as word order and synonym matching, to arrive at a comprehensive assessment of translation quality. The formal definition of METEOR is as follows:

$$METEOR = \frac{10PR}{R + 9P}(1 - Penalty)$$

The penalty factor accounts for word order and length differences between the hypothesis and reference.

Perplexity: Perplexity, a widely used metric for evaluating autoregressive or causal language models, quantifies the degree of uncertainty a model exhibits when predicting the next token in a sequence. It is formally defined as the exponential average negative log-likelihood of a given text sequence. Mathematically, for a text sequence X of length m generated using a model $f_\theta(.)$, perplexity can be calculated as:

$$ppl(X) = exp\left(-\frac{1}{m}\sum_{i}^{m} \log f_\theta(x_i|x_{<i})\right)$$

where, $f_\theta(x_i|x_{<i})$ signifies the probability assigned by the model to the token x_i, conditioned on the preceding tokens $x_{<i}$.

C.3 Metrics for Nutrition Generation

Micro-nutrients were formatted in a pre-defined JSON style during the training, so the model was also expected to generate text in a similar style, making it easy to parse micro-nutrients and their values. Note that each generated sample output may not contain all the desired micro-nutrients. Therefore, for each sample, we consider the micro-nutrient tags found in the generated text. We parse all the micro-nutrient tags from the generated text along with their numerical values and compute the mean average error with the ground truth, formally defined as:

$$MAE(nutri) = \frac{1}{n}\sum_{i=1}^{n} |y^i_{nutri} - \tilde{y}^i_{nutri}|$$

Where, y^i_{nutri} and $\tilde{y}^i_{nutri}$ are the ground truth and predicted values of the nutrient, respectively.

D Additional Results

D.1 *KERL-Recom*

Performance on Recipe Types Table 13 shows the performance of open source LLMs and our model on the KGQA benchmark for recipes tagged with some of the tags such as lactose, vegan, vegetarian, gluten-free and nut-free. LLaMA-2 ranks as the second best, except for the gluten-free tag. Mistral performs similarly to or slightly better than Phi-3-mini, but Phi-3-mini is smaller than the other models. Overall, *KERL-Recom*, leveraging Phi-3-mini as the base model, achieves precision and F1 greater than 90 for all tags.

Qualitative results Despite the impressive performance of *KERL-Recom*, it is prone to failure by recommending false positives or missing true positive in recommended recipes. For examples,

Model	Inputs	Calories	Fat Calories	Protein	Sugar	Fiber	Carbohydrates	Sodium	Cholesterol	Saturated Fat	Total Fat
Dataset Mean		321.76 ± 222.82	135.36±122.82	12.69± 12.51	10.01± 11.89	2.67± 2.54	29.86±23.94	0.44±0.44	0.05±0.06	5.62±5.74	15.09 ± 13.65
LLaVA-Chef	X_t	205.13	118.21	9.22	9.6	3.07	20.9	28.66	2.69	8.07	17.97
LLaVA-Chef	X_{ing}	229.17	111.5	13.23	14.61	6.82	27.45	155.73	15.36	18.13	34.44
LLaVA-Chef	X_{inst}	222.32	111.05	13.25	15.34	7.23	27.38	110.52	13.5	17.94	32.85
LLaVA-Chef	$X_t + X_{ing} + X_{inst}$	233.35	113.36	10.5	14.01	4.81	28.4	188.49	19.37	17.79	36.52
KERL-Nutri	X_t	159.44	85.58	6.64	6.62	1.61	16.17	0.29	0.03	3.91	9.51
KERL-Nutri	X_{ing}	132.62	61.23	5.47	4.82	1.22	12.62	0.21	**0.02**	2.56	6.8
KERL-Nutri	X_{inst}	147.19	75.84	6.29	6.11	1.46	14.91	0.26	0.03	3.27	8.43
KERL-Nutri	$X_t + X_{ing} + X_{inst}$	**127.67**	**59.49**	**5.3**	**4.64**	**1.18**	**12.09**	**0.2**	**0.02**	**2.48**	**6.61**

Table 12: Performance of nutrition generation models, filtered to include only the 95th percentile of samples.

Model	Tag	mAP	P	R	F1
internLM2	lactose	0.015	0.008	0.017	0.011
Mistral		0.14	0.538	0.368	0.437
Llama-2		0.477	0.838	0.514	0.637
Llama-3.1		0.152	0.233	0.321	0.27
Phi-3-mini-128K		0.202	0.812	0.217	0.343
KERL-Nutri		**0.898**	**0.955**	**0.916**	**0.935**
internLM2	vegan	0.09	0.038	0.0079	0.051
Mistral		0.201	0.549	0.492	0.519
Llama-2		0.669	0.885	0.71	0.788
Llama-3.1		0.161	0.296	0.43	0.351
Phi-3-mini-128K		0.404	0.873	0.421	0.568
KERL-Nutri		**0.964**	**0.988**	**0.975**	**0.981**
internLM2	vegetarian	0.085	0.034	0.078	0.048
Mistral		0.201	0.531	0.52	0.526
Llama-2		0.639	0.856	0.708	0.775
Llama-3.1		0.179	0.325	0.462	0.381
Phi-3-mini-128K		0.361	0.871	0.352	0.501
KERL-Nutri		**0.966**	**0.987**	**0.976**	**0.981**
internLM2	gluten-free	0.062	0.027	0.058	0.037
Mistral*		0.26	0.581	0.673	0.624
Llama-2		0.559	0.87	0.609	0.717
Llama-3.1		0.208	0.364	0.548	0.438
Phi-3-mini-128K		0.282	0.811	0.285	0.421
KERL-Nutri		**0.939**	**0.982**	**0.951**	**0.966**
internLM2	nut-free	0.103	0.019	0.067	0.029
Mistral*		0.224	0.542	0.591	0.565
Llama-2		0.628	0.833	0.682	0.75
Llama-3.1		0.243	0.345	0.455	0.392
Phi-3-mini-128K		0.385	0.786	0.367	0.5
KERL-Nutri		**1.0**	**0.909**	**1.0**	**0.952**

Table 13: Results on KGQA test set reported for several tags. Overall, *KERL-Recom* performs better for numerous types of recipes.

row-2 in Table 14 shows the question where *KERL-Recom* suggested false positive whereas in row-3 it suggested a recipe that is not even in context. Similarly, the last row demonstrates an example, where the model failed to select all true positives from context, resulting few true negatives.

D.2 *KERL-Nutri*

In Table 9, we compare the performance of LLaVA-Chef and our *KERL-Nutri* model on generating the micro-nutrients for the recommended recipes. For some nutrients, the MAE is rather large. This happens because some ground truth samples have abnormally high or zero nutritional values, introducing noise that affects model performance. To analyze this, we filtered the samples within a specific percentile range, excluding outliers, and then calculated the MAE. Detailed results on nutrient generation for the 95th percentile of samples are shown in Table 12. We observe that removing noisy samples reduces MAE by about half for all the nutrients. Nevertheless, our *KERL-Nutri* remains the superior model.

User question	Recipe in context	*KERL-Recom* recommendations
What low-protein recipes use crushed red pepper flakes, bacon, ginger ale, ground black pepper, pepper and exclude cream of coconut, tamari, fresh thyme leaves, and have fiber more than 4.28, sugars per 100g within range (0, 5.99)?	Tamarind Juice, Mock Sangria, Low-Carb Balsamic Dressing, County Cherry Dessert, Mock Champagne	Low Carb Balsamic Dressing, Mock Champagne, Mock Sangria
Suggest me vegetarian dishes that require fresh ground black pepper, green onions, fresh parsley, plain yogurt, red onions and must not contain fine salt, peach slices, arhar dal, and have a total of fiber not above 5.4, sugars per 100g no more than 3.66.	Gyoza or Pot Sticker Dipping Sauce, Wild Rice Stuffing Side Dish, Pixie Cookies	Wild Rice Stuffing Side Dish, Pixie Cookies
Can you list the low-carb recipes that use curry powder, cooked spaghetti, steak, bottled hot pepper sauce, condensed beef broth but do not contain whole wheat pancake mix, chicken thigh fillets, white bread, while containing fat cals not less than 203.0, and protein within range (0, 32.5)?	Quick Sausage, White Bean and Spinach Stew, Jamaican Brown Stew Chicken, Watermelon, Cucumber and Feta Salad	Jamaican Brown Stew Chicken, Manic Bullet
Can you suggest low-sodium recipes cooked with fresh ground black pepper, plain yogurt, all - purpose flour, fresh lemon juice, apples but do not have garam masala, wheat and have protein no more than 13.08, sugars per 100g no more than 11.53?	Fudge Pie, Pasta Pascal, Chocolate-Pecan Brownies, Cold Oven Pound Cake, Cucumber and Feta Salad, Lemon Meringue Tart, Cold Oven Pound Cake	Fudge Pie, Cold Oven Pound Cake, Lemon Meringue Tart

Table 14: Qualitative results of KERL-Recom: The second column lists the recipes in the subgraph (only names for simplicity) where blue color shows recipes that satisfy the user constraints $R^+(t_j)$. Row 1 shows a perfect result, row 2 shows one false positive recommendation, row 3 shows two suggested recipes, with only one present in the context and is also true positive, and the final row shows a subset of the ground truth selected by the model with missing true positives from recommended recipes. These sample results suggest that despite showing strong performance, it may fail by suggesting false negatives, missing true positives, or recommending unrelated recipes.