Food-Energy-Water Nexus
WISDOM ACADEMIC https://doi.org/10.63808/fewn.v2i3.428 Volume 2 , Issue 3

Research Article

A Constraint-Aware Knowledge Graph RAG Framework for Personalized Food Recommendation

Jia Yu[1,*], Tong Ji[1], Jiamin Zheng[1]
[1]School of Computer and Artificial Intelligence, Beijing Technology and Business University, Beijing 102488, China
[*]**Correspondence:** Jia Yu, 3014529637@qq.com

© The Author(s) 2026.
This article is published by Wisdom Academic Press Ltd. under a Creative Commons Attribution 4.0 International License. The license permits use, sharing, adaptation, distribution, and reproduction in any medium or format, provided that appropriate credit is given to the original author(s) and the source, a link to the license is supplied, and any modifications are indicated. Unless otherwise stated, all images and third-party materials included in this article are also covered by the same license. For any material not covered by this license, if the intended use is neither permitted by applicable law nor allowed under the license terms, direct permission must be obtained from the copyright holder. To view a copy of this license, please visit http://creativecommons.org/licenses/by/4.0/.

Abstract

To address the issues of knowledge hallucination, insufficient constraint satisfaction, and opaque recommendation rationale in Large Language Models (LLMs) during food recommendation tasks, this paper proposes a personalized food recommendation method based on Constraint-Aware Knowledge Graph Retrieval-Augmented Generation (RAG). First, the method performs intent recognition and constraint extraction on users' natural language requirements, transforming meal types, calorie limits, nutritional goals, allergy contraindications, and user preferences into structured query conditions. Subsequently, it retrieves relevant candidate dishes, ingredients, and nutritional subgraphs within the food knowledge graph. On this basis, a recommendation ranking function is designed, incorporating constraint matching, nutritional balance, user preferences, graph path relevance, and risk penalties. Finally, the ranked results and evidence subgraphs are input into the LLM as retrieval-augmented context to generate interpretable meal suggestions. Experimental results show that, compared with existing methods, the proposed method achieves superior performance in recommendation accuracy, constraint satisfaction rate, and response faithfulness. Furthermore, sensitivity analyses, performance evaluations and case studies provide further evidence that constraint-aware knowledge graph RAG can significantly enhance the reliability and explainability of food recommendation.

Keywords

Food Recommendation, Knowledge Graph, Retrieval-Augmented Generation, Large Language Models, Constraint Satisfaction

1. Introduction

The enhancement of residents' health awareness has propelled the development of personalized dietary recommendation technologies (e.g., Ceccaldi et al., 2022; Misra et al., 2020). Unlike general product recommendation, food meal planning must not only cater to users' taste preferences but also simultaneously satisfy multiple constraints, including calorie control, nutritional balance, allergy contraindications, disease risks, and situational requirements. Users typically express their needs in natural language, such as, "I want a low-calorie, high-protein, dairy-free lunch." Such requirements are characterized by semantic ambiguity, complex constraint combinations, and a high demand for explainability, making it difficult for traditional recommendation methods based on collaborative filtering or keyword matching to fully meet these needs.

Large Language Models (LLMs) possess strong natural language understanding and generation capabilities (Brown et al., 2020; Dettmers et al., 2023; Hu et al., 2022; Vaswani et al., 2017), enabling them to produce well-structured dietary suggestions. However, directly applying LLMs to meal planning presents challenges such as factual inadequacy, unclear nutritional rationale, and unstable constraint satisfaction. Retrieval-Augmented Generation (RAG) alleviates knowledge hallucination to a certain extent by incorporating external knowledge sources to provide factual grounding for the model (Borgeaud et al., 2022; Lewis et al., 2020; Nakano et al., 2021; Wang et al., 2025). Nevertheless, standard vector-based RAG relies primarily on text similarity for retrieval, making it difficult to explicitly handle structured constraints like "under 400 kcal", "peanut-free", or "suitable for blood sugar control."

Knowledge Graphs (KGs) organize food and meal planning knowledge as entities, relations, and attributes, and are therefore suitable for representing relationships among ingredients, dishes, nutrients, target populations, and dietary taboos (Hogan et al., 2021; Ma et al., 2024; Min et al., 2022; Pan et al., 2024; Peng et al., 2023; Yang et al., 2025). Integrating KGs with RAG allows the generation process of LLMs to be grounded in structured retrieval results and interpretable paths, thereby enhancing the reliability of meal recommendations.

As the current dataset lacks reliable user-level labels for cooking energy consumption and water footprint, the experiments in this paper remain focused on food nutrition, safety constraints and interpretable recommendations; this limitation, along with potential avenues for future expansion, is clearly outlined in the Discussion and Limitations section. The framework proposed in this paper can serve as a decision-support tool for food nutrition and has the potential to be extended to the dimensions of energy and water resources. Specifically, cooking energy consumption can be modelled through relationships such as 'dish-requires-cooking method' and 'cooking method-consumes-energy', whilst the water

footprint of ingredients can be modelled through the relationship 'ingredient-has-water footprint'. As the current dataset lacks reliable user-level labels for cooking energy consumption and water footprints, this study remains focused on food nutrition, safety constraints and interpretable recommendations, and clearly outlines this limitation and future extension pathways in the discussion and limitations section.

To address these issues, this paper proposes a personalized food meal planning recommendation method based on Constraint-Aware Knowledge Graph RAG. The main contributions of this paper are as follows: First, we propose a user requirement parsing method for meal planning that transforms natural language input into structured constraints. Second, we design a KG subgraph retrieval strategy to acquire candidate dishes and evidence paths relevant to user needs. Third, we introduce a constraint-aware recommendation ranking function that comprehensively considers constraint matching, nutritional balance, preferences, graph path relevance, and risk penalties. Fourth, we construct a RAG generation template that inputs evidence subgraphs into the LLM to achieve interpretable meal planning recommendations.

2. Related Work

2.1. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) enhances a model's ability to utilize factual knowledge by incorporating external retrieval results into the input of generative models. The RAG framework proposed by Lewis et al. (2020) combines parametric language models with non-parametric knowledge retrieval for knowledge-intensive natural language processing tasks. RETRO enhances language model performance by retrieving relevant texts from large-scale external databases (Borgeaud et al., 2022), while WebGPT improves question-answering quality by combining web browsing with human feedback (Nakano et al., 2021). These studies demonstrate that retrieval mechanisms can effectively mitigate issues of outdated knowledge and hallucination in LLMs.

However, in the context of food meal planning, user requirements often involve numerical and logical constraints. Relying solely on text similarity retrieval tends to yield results that fail to meet these conditions. For instance, vector retrieval might locate texts related to "high-protein recipes" but fail to guarantee that the total calories remain below the user-specified threshold. Therefore, it is essential to introduce structured knowledge retrieval and constraint verification mechanisms into RAG (Bao et al., 2016).

2.2. Knowledge Graphs and Food Recommendation

Knowledge Graphs (KGs) are commonly used in recommender systems to model multi-hop relationships among users, items, and attributes. Methods such as KGAT learn high-order correlations between users and items via Knowledge Graph Attention Networks (Wang et al., 2019). In the food domain, KGs can connect dishes, ingredients, nutrients, dietary restrictions, and health goals, providing structured support for nutritional analysis, food safety, and personalized recommendation (Min et al., 2022). The health-aware food recommendation model FKGM (Chen et al., 2023), which is based on knowledge graphs and multi-task learning, and the NR-KGNN (Ma et al., 2024) method, which is based on nutrition-related knowledge graphs and GCNs, represent the health-aware multi-task learning and nutrition knowledge graph neural network paradigms respectively, and are representative external baseline methods in the field of food recommendation.

Food recommendation differs from general recommendation tasks in that its outcomes have direct health implications, thus requiring higher accuracy and explainability. Beyond user preferences, the system must also identify allergens, disease-related contraindications, and nutritional intake goals. Paths within a KG can directly illustrate the rationale behind a recommendation--for example, dish-contains-ingredient-rich in-nutrient-contributes to-health goal--providing a natural foundation for interpretable outputs.

2.3. Constraint-Aware Recommendation

Constraint-aware recommendation emphasizes the explicit consideration of user-specified conditions during the recommendation process. Traditional constraint-based recommendation often employs filtering and rule matching to eliminate candidates that do not meet the criteria (Felfernig et al., 2015). For food meal planning tasks, constraints include not only hard constraints, such as allergy contraindications and calorie limits, but also soft constraints, such as taste preferences, cooking complexity, and the degree of nutritional balance. Relying solely on hard filtering can lead to an overly small candidate set, whereas relying only on similarity ranking may overlook safety risks.

This paper categorizes user constraints into hard and soft constraints. Hard constraints are primarily used for candidate filtering and risk penalisation, whilst soft constraints are employed for ranking weighting to evaluate the effectiveness of different constraint handling strategies. The experiments include two types of baseline methods: the traditional ‘rule-based filtering + nutritional scoring’ approach and a low-sample prompt-engineering LLM method. This allows for a direct comparison of the performance differences between rule-based constraint-driven recommendations, prompt-based LLM recommendations, and the

constraint-aware KG-RAG method proposed in this paper in the context of food meal planning tasks.

2.4. Mechanisms for Extending Sustainable Dietary Recommendations

Food recommendations not only influence nutritional intake but may also affect energy consumption during cooking and the water footprint of food production. Different dishes require different cooking methods, cooking times and types of equipment, thereby resulting in varying energy requirements; there are also significant differences in the water footprints of different ingredients (Hoekstra & Chapagain, 2008). Therefore, this paper reserves slots for the expansion of the knowledge graph to accommodate cooking energy consumption and ingredient water footprints, laying the foundation for subsequent recommendation models that simultaneously optimise nutritional, energy and water sustainability.

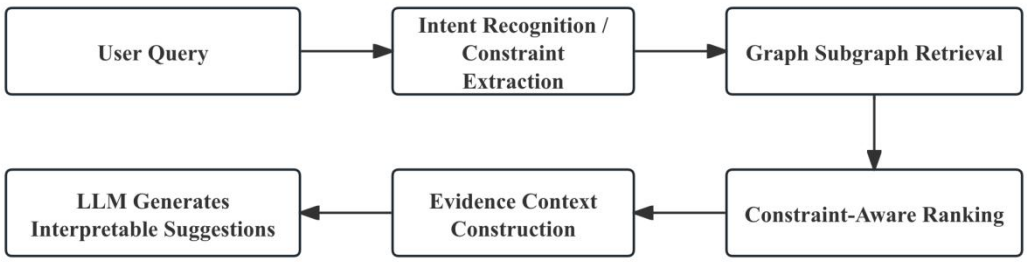
3. Methods

3.1. Overall Framework

The KG-CRAG (Knowledge Graph-based Constraint-aware Retrieval-Augmented Generation) method proposed in this paper consists of six steps: requirement parsing, structured query construction, knowledge graph subgraph retrieval, candidate ranking, evidence organization, and augmented generation. The system first identifies user intent and extracts constraints, then retrieves candidate dishes and evidence paths that satisfy the conditions from the food knowledge graph. Subsequently, it ranks the candidates using a constraint-aware scoring function and finally inputs the retrieval results into an LLM to generate meal planning suggestions.

Figure 1

Overall workflow of the KG-CRAG method



Note. Workflow of the proposed Constraint-Aware Knowledge Graph Retrieval-Augmented Generation (KG-CRAG) framework for personalized food recommendation.

3.2. Food Knowledge Graph Construction, Quality Control and Availability

The food knowledge graph is represented as G=(E,R,A,T), where E is the entity set, R is the relation set, A is the attribute set, and T is the set of triples. The knowledge graph construction process is as follows: as shown in Table 1, nutrient attributes are normalized mainly from USDA FoodData Central (U.S. Department of Agriculture, n.d.), recipe-level dish and ingredient records are manually curated from public recipe descriptions, allergen and dietary restriction labels are compiled from common food safety categories, and health-goal mappings are curated from nutrition rules used by dietitians. The current KG contains 1,200 dish entities, 2,600 ingredient entities, 30 nutrient attributes, 14 relation types, and 18,420 triples. Each nutrient value is converted to a per-serving basis and checked by unit consistency rules. Duplicate ingredients are merged through synonym normalization, and 10% of triples are sampled for manual inspection; the observed triple-level precision is 96.7%.

Table 1

Knowledge graph schema and data sources

Component	Entity/Relation examples	Main data source	Quality-control rule
Dish	Chicken breast quinoa salad, tofu soup	Curated recipe records	Duplicate-name merging and serving-size normalization
Ingredient	Chicken breast, quinoa, tofu, peanut	Recipe ingredient lists+USDA FDC mapping	Synonym dictionary and manual spot check
Nutrient	Energy, protein, fat, carbohydrate, fiber, sodium, sugar	USDA FoodData Central	Unit conversion to per serving and range check
Allergen/restriction	Peanut, dairy, shellfish, gluten, vegetarian	Food safety labels and manual rules	Hard-constraint validation by relation paths
Health goal	Fat loss, blood sugar control, muscle gain, cardiovascular health	Nutrition rule base	Expert-rule consistency check
FEW extension slot	Cooking method, cooking time, water footprint	To be linked with energy/water datasets	Currently stored as optional attributes

Note. FEW = Food-Energy-Water; USDA FDC = U.S. Department of Agriculture FoodData Central.

Table 2

Representative examples of knowledge graphs

Head entity	Relation	Tail entity / attribute	Example value
Chicken breast quinoa salad	contains_ingredient	Chicken breast	-

Chicken breast	rich_in	Protein	31 g/100 g
Chicken breast quinoa salad	has_calorie	Energy	360 kcal/serving
Chicken breast quinoa salad	excludes_allergen	Dairy	true
Peanut sauce noodles	contains_allergen	Peanut	true
Stir-fried tofu	requires_cooking_method	Stir-frying	optional FEW extension
Almond	has_water_footprint	Water footprint	reserved attribute

Note. Selected representative examples from the custom food knowledge graph.

Due to copyright restrictions on some curated recipe descriptions, the full KG cannot be publicly released in its raw form. Table 2 provides only a selection of anonymised representative examples.

3.3. User Requirement Parsing

User input is typically in the form of natural language text, denoted as q. This paper defines the requirement parsing task as extracting the intent I and the constraint set C from q. Intent types include recipe search, nutritional consultation, contraindication assessment, substitution recommendation, and meal combination planning. The constraint set C encompasses meal types, calorie ranges, macronutrient goals, dietary patterns, allergy contraindications, disease-related restrictions, and taste preferences.

Table 3

User constraint types

Constraint category	Attribute examples	Constraint property
Basic scenario	Breakfast, lunch, dinner, snack	Soft constraint
Nutrition target	Low calorie, high protein, low fat, low sugar	Soft/hard constraint
Contraindication risk	Peanut allergy, lactose intolerance, seafood allergy	Hard constraint
Health goal	Fat loss, blood sugar control, muscle gain, cardiovascular health	Soft constraint
User preference	Light taste, low oil, vegetarian, flavor preference	Soft constraint
FEW extension	Low cooking energy, low water footprint	Reserved soft constraint

Note. Classification of user requirement constraints in the meal planning parser.

To minimize parsing errors, this paper adopts a hybrid approach combining a rule-based dictionary with LLM correction. The rule-based dictionary identifies common nutritional and contraindication expressions, while the LLM handles complex semantics and elliptical

expressions. The parser outputs a JSON-style structure, e.g., {meal=lunch, calorie<=400, protein=high, dairy=prohibited}. Hard constraints are passed to the filtering stage, whereas soft constraints are passed to ranking. For ambiguous phrases, the system maps them to predefined thresholds in Table 4; for instance, high protein is defined as at least 20 g protein per serving.

Table 4

Operational thresholds for common soft constraints

Soft constraint	Operational threshold in this study
Low calorie	<=400 kcal/serving for lunch or dinner; <=250 kcal/serving for snack
High protein	>=20 g protein/serving or >=20% of total energy from protein
Low fat	<=15 g fat/serving
Low sugar	<=10 g sugar/serving
Low oil/light taste	<=10 g added oil/serving and sodium <=600 mg/serving
Vegetarian	No meat, fish, seafood, or animal-derived broth

Note. Operational definition of quantitative and qualitative soft constraints used in this study.

3.4. Knowledge Graph Subgraph Retrieval

For the structured constraints C, the system first filters candidate dishes based on hard constraints. For example, if a user has a peanut allergy constraint, all dishes containing peanut or nut allergen paths are eliminated. Subsequently, the system retrieves k-hop subgraphs relevant to the target based on soft constraints.

Given a candidate dish d_i, its evidence subgraph S_i consists of paths related to ingredients, nutritional attributes, and health goals associated with d_i:

$$S_i=\{p|p \text{ starts from } d_i \text{ and length}(p)\leq k, p \text{ satisfies } C\} \quad (1)$$

where p denotes a graph path, and k is the maximum path length. In our experiments, k is set to 3 to balance explanatory completeness and retrieval efficiency.For each query, the KG retrieval module returns at most 50 candidate dishes before ranking, which avoids excessive context length in the LLM generation stage.

3.5. Constraint-Aware Recommendation Ranking

After obtaining the candidate dishes, this paper designs a constraint-aware recommendation scoring function:

$$\text{Score}(q,d_i)=\lambda_1 C_{\text{match}}+\lambda_2 N_{\text{score}}+\lambda_3 P_{\text{pref}}+\lambda_4 G_{\text{path}}-\lambda_5 R_{\text{risk}} \quad (2)$$

Food-Energy-Water Nexus
WISDOM ACADEMIC https://doi.org/10.63808/fewn.v2i3.428 Volume 2 , Issue 3

All sub-scores are normalized to [0,1]. Let H_q and S_q denote hard and soft constraints, respectively. A candidate is removed if any h in H_q is explicitly violated. For the remaining candidates, C_{match} is the weighted proportion of satisfied constraints:

$$C_{match}=\frac{\sum_{c\in C_q} w_c \cdot I[d_i \ satisfies \ c]}{\sum_{c\in C_q} w_c} \tag{3}$$

where $C_q=H_q \cup S_q$ denotes the union of all hard and soft constraints; w_c is the weight of constraint c; and $I[\cdot]$ is the indicator function, which takes the value 1 if the condition within the brackets is satisfied, and 0 otherwise. The nutritional balance score is:

$$N_{score}=1-\frac{1}{|M_q|}\sum_{m\in M_q}\min\left(\frac{|x_{i,m}-t_{q,m}|}{r_m},1\right) \tag{4}$$

where $x_{i,m}$ is the nutrient value of dish d_i, $t_{q,m}$ is the target or threshold value, and r_m is the acceptable range. The preference score P_{pref} is computed as the cosine similarity between the user preference vector and the dish attribute vector. The formula for graph-path relevance score is as follows:

$$G_{path}=\min\left(1,\ \sum_{p\in P(d_i,C_q)} \gamma^{length(p)-1}\cdot \alpha(p)\right) \tag{5}$$

where $\gamma=0.8$ is a path-length discount factor and $\alpha(p)$ is relation reliability. The risk penalty is defined as:

$$R_{risk}=\sum_{r\in R_q}\rho_r \cdot I[r \ is violated \ or \ unknown] \tag{6}$$

where severe allergen risks use $\rho_r=1$ and unknown mild-risk attributes use $\rho_r=0.3$.

Table 5

Components of the recommendation scoring function

Symbol	Meaning	Calculation basis
C_{match}	Constraint matching score	Weighted satisfaction of calorie, meal type, diet pattern, and allergy constraints
N_{score}	Nutritional balance score	Deviation from calorie and macronutrient targets after normalization
P_{pref}	User preference score	Similarity between user taste/cooking preferences and dish attributes
G_{path}	Graph path relevance	Discounted reliable paths between candidate dishes and target entities
R_{risk}	Risk penalty	Allergy, contraindication, calorie-excess and unsuitable-population risks

Note. Components and calculation rationales of the overall constraint-aware ranking score function.

The default weights are $\lambda_1=0.30$, $\lambda_2=0.20$, $\lambda_3=0.15$, $\lambda_4=0.20$, and $\lambda_5=0.15$. These weights were selected on a validation subset because they jointly maximize Acc and Hit@3

while keeping CSR above 92%; the sensitivity analysis in Section 4.6 reports the effect of varying the weights.

3.6. RAG-Augmented Generation and Interpretable Output

Following the ranking process, the system organizes the Top-k candidate dishes and their evidence subgraphs as context to input into the Large Language Model. The context includes dish names, main ingredients, key nutritional attributes, constraint satisfaction status, and graph evidence paths. The generation template requires the model to answer strictly based on the retrieved results and to output recommendation reasons, precautions, and alternative options.

For instance, for the query “Recommend a low-calorie, high-protein, dairy-free lunch”, the system retrieves the candidate dish “Grilled Chicken Quinoa Salad”, with evidence paths including “Chicken breast - rich in - Protein”, “Quinoa - rich in - Dietary Fiber”, “This dish - has calorie count - 360 kcal” and “This dish - contains no - Dairy”. When generating the final recommendation rationale, the model must cite these pieces of evidence, thereby reducing the risk of hallucination.

3.7. Algorithm Description

Table 6

Steps of the KG-CBAG Algorithm

Step	Operation
Input	User query q , Food Knowledge Graph G, User Profile U, Number of candidate returns Top-k
1	Identify user intent I, extract constraint set C
2	Preliminarily filter candidate dishes in the knowledge graph based on hard constraints
3	Retrieve k-hop evidence subgraphs S_i around candidate dishes
4	Calculate C_{match} , N_{score}, P_{pref}, G_{path} and R_{risk}
5	Rank candidate dishes according to $Score(q,d_i)$, and select Top-k results
6	Construct RAG context and invoke a Large Language Model to generate explainable meal planning suggestions
Output	Meal planning recommendation results, recommendation reasons, and evidence paths

Note. Algorithm procedure for the proposed KG-CRAG framework.

4. Experiments and Analysis of Results

4.1. Datasets and Experimental Setup

This paper constructs the experimental data based on a food meal planning knowledge graph, which includes 1,200 candidate dishes, 2,600 ingredient entities, 30 types of nutritional attributes, and common dietary restriction relationships. The user query test set contains 200 natural language meal planning requests, covering scenarios such as low-calorie, high-protein, vegetarian, allergy avoidance, blood sugar control, and personalized preferences. Each query was independently annotated by three annotators using a written guideline. The guideline requires annotators to mark all explicit constraints, judge whether the top recommendation is acceptable, and verify whether each generated nutritional or ingredient claim is supported by the retrieved KG evidence. Disagreements were resolved by majority voting. Fleiss' Kappa is 0.82 for Acc labels and 0.79 for Faithfulness labels, indicating substantial agreement. In the experiments, Top-k is set to 3, the maximum retrieval path length k is set to 3, and λ_1, λ_2, λ_3, λ_4, and λ_5 are set to 0.30, 0.20, 0.15, 0.20, and 0.15, respectively.

The computational environment is specified as follows: Ubuntu 22.04, Python 3.10, PyTorch 2.2, Transformers 4.41, sentence-transformers 2.7, FAISS 1.8, Neo4j 5.18, CUDA 12.1, Intel Xeon Silver CPU, 128 GB RAM, and one NVIDIA RTX 4090 GPU with 24 GB memory. Vector retrieval uses BAAI/bge-m3 embeddings. ART is measured on a single-query online inference setting with batch size 1 for generation and batch size 16 for embedding retrieval precomputation.

Table 7

Distribution of user query test set types

Query type	Count	Example
Low-calorie control	42	Recommend a dinner under 400 kcal
High-protein requirement	38	Want a high-protein lunch suitable for post-workout
Allergy/contraindication avoidance	36	No peanuts and dairy products
Health goals	44	Meal plans suitable for fat loss or blood sugar control
Comprehensive preferences	40	Light, low-oil lunch suitable for vegetarians

Note. Category distribution and representative query examples from the evaluation dataset (N = 200).

4.2. Evaluation Metrics

This paper adopts recommendation accuracy (Acc), constraint satisfaction rate (CSR), Hit@3, answer faithfulness (Faithfulness), and average response time (ART) as evaluation metrics. The formulas for each indicator are as follows:

$$\text{Acc}=\frac{1}{N}\sum_{i} \text{I (the top}-1 \text{ recommendation is acceptable)} \quad (7)$$

$$\text{CSR}=\frac{1}{N}\sum_{i} \frac{\text{satisfied explicit constraints}}{\text{explicit constraints}} \quad (8)$$

$$\text{Hit@3}=\frac{1}{N}\sum_{i} \text{I (at least one acceptable recommendation appears in the top three)} \quad (9)$$

$$\text{Faithfulness}=\frac{1}{N}\sum_{i} \text{I (all nutritional,allergen,and ingredient claims are supported by retrieved evidence)} \quad (10)$$

ART is the average wall-clock latency per query. For human-judged metrics, method names were hidden from annotators, and each method produced 200 top-1 generated answers and 600 top-3 candidate items for evaluation.

All randomized generation experiments are repeated with five random seeds (0-4). For deterministic retrieval baselines, we use 1,000 bootstrap resamples over the 200-query test set. Results are reported as mean +/- standard deviation, and paired two-sided Wilcoxon signed-rank tests are conducted between KG-CRAG and each baseline.

4.3. Comparison Experiments and Statistical Significance

This paper compares the performance of the proposed KG-CRAG method with that of various baseline methods—including LLM-Direct, LLM-Prompt, Rule-Nutrition, Vector-RAG, DenseRetriever-RAG, KGAT, FKGM, NR-KGNN and KG-RAG—on the meal recommendation task; specific implementation details are shown in Table 8.

Table 8

Baseline methods and implementation details

Method	Implementation details
LLM-Direct	Qwen2.5-7B-Instruct directly generates recommendations from the user query without retrieval.
LLM-Prompt	Few-shot prompt with three examples and explicit constraint instructions, but without external retrieval.
Rule-Nutrition	Traditional constraint-based recommender: hard-rule filtering plus nutrition score ranking.
Vector-RAG	BAAI/bge-m3 retrieves top recipe texts; Qwen2.5-7B-Instruct generates with retrieved text.
DenseRetriever-RAG	Dense retriever fine-tuned on 160 labeled queries; generation uses the same LLM template.
KGAT	Knowledge graph attention network adapted to dish-ingredient-nutrient-user preference graph.
FKGM	Health-aware food recommendation based on KG and multi-task learning, adapted from Chen et al. (2023).
NR-KGNN	Nutrition-related KG neural network based on GCN, adapted from Ma et al. (2024).

KG-RAG	KG entity/relation retrieval without constraint-aware ranking or risk penalty.
KG-CRAG (Ours)	Hard filtering, KG subgraph retrieval, constraint-aware ranking, risk penalty, and evidence-grounded LLM generation.

Note. Descriptions and baseline setups for performance comparison.

The experimental results are shown in Table 9, compared with the strongest external dietary recommendation baseline (NR-KGNN), KG-CRAG improves Acc by 3.5 percentage points, CSR by 6.5 percentage points, and Hit@3 by 4.0 percentage points. Paired Wilcoxon tests show that the improvements over NR-KGNN are significant for Acc (p=0.018), CSR (p=0.009), Hit@3 (p=0.021), and Faithfulness (p=0.017). Improvements over KG-RAG are also significant (Acc p=0.026, CSR p=0.011, Hit@3 p=0.034, Faithfulness p=0.028). These results indicate that the gain is not only due to KG retrieval, but also to explicit constraint-aware ranking and risk penalization.

Table 9

Comparison of recommendation performance across methods (%, mean +/- standard deviation)

Method	Acc	CSR	Hit@3	Faithfulness	ART/s
LLM-Direct	70.5 +/- 2.4	65.2 +/- 3.1	73.0 +/- 2.6	68.0 +/- 2.8	2.18
LLM-Prompt	75.6 +/- 2.1	78.4 +/- 2.4	80.0 +/- 2.1	75.2 +/- 2.5	2.34
Rule-Nutrition	80.1 +/- 1.8	88.6 +/- 1.7	85.5 +/- 2.0	79.0 +/- 1.9	1.45
Vector-RAG	78.0 +/- 2.2	76.3 +/- 2.6	82.5 +/- 1.9	81.7 +/- 2.2	2.86
DenseRetriever-RAG	82.8 +/- 1.9	80.6 +/- 2.1	86.2 +/- 1.8	83.4 +/- 2.0	2.94
KGAT	83.1 +/- 1.7	81.4 +/- 2.0	86.7 +/- 1.7	82.1 +/- 1.8	1.83
FKGM	85.2 +/- 1.6	84.1 +/- 1.8	88.4 +/- 1.5	84.7 +/- 1.7	2.06
NR-KGNN	86.0 +/- 1.5	85.6 +/- 1.7	89.0 +/- 1.4	85.9 +/- 1.6	2.11
KG-RAG	84.5 +/- 1.8	86.4 +/- 1.6	88.0 +/- 1.5	87.3 +/- 1.6	2.52
KG-CRAG (Ours)	89.5 +/- 1.3	92.1 +/- 1.0	93.0 +/- 1.2	91.5 +/- 1.1	2.73

Note. Values are presented as mean ± standard deviation. Acc = Accuracy; CSR = Constraint Satisfaction Rate; ART = Average Response Time. All paired Wilcoxon signed-rank tests against KG-CRAG were statistically significant ($p < 0.05$).

4.4. Ablation Experiment

Table 10

Ablation experiment results (%, mean)

Method	Acc	CSR	Hit@3	Faithfulness
Full KG-CRAG	89.5	92.1	93.0	91.5
Remove C_{match}	82.4	78.9	86.0	87.5
Remove N_{score}	87.4	90.8	91.6	90.9
Remove P_{pref}	87.0	90.1	91.2	90.4
Remove G_{path}	86.0	89.2	90.4	88.6
Remove R_{risk}	83.5	80.4	87.1	86.1
Remove KG retrieval	78.0	76.3	82.5	81.7

Note. Ablation study performance scores across different evaluation metrics.

The results of the ablation experiments are shown in Table 10. The experiments indicate that C_{match} and R_{risk} have the greatest impact on constraint satisfaction. After removing R_{risk}, CSR drops to 80.4%, suggesting that risks such as allergies, contraindications, and calorie excess cannot rely solely on the generative model's self-judgment. Removing G_{path} leads to a decrease in answer faithfulness, indicating that graph paths help provide explainable evidence. After removing user preferences, the overall accuracy decreases, demonstrating that personalized information can improve candidate ranking.

4.5. Sensitivity Analysis on Weight Parameters

Table 11

Sensitivity analysis of λ_1- λ_5

Setting	(λ_1, λ_2, λ_3, λ_4, λ_5)	Acc	CSR	Hit@3
Balanced default	(0.30, 0.20, 0.15, 0.20, 0.15)	89.5	92.1	93.0
Constraint-heavy	(0.40, 0.15, 0.10, 0.15, 0.20)	89.0	93.4	92.6
Nutrition-heavy	(0.25, 0.30, 0.10, 0.20, 0.15)	88.8	91.6	92.4
Preference-heavy	(0.25, 0.15, 0.30, 0.15, 0.15)	87.9	89.7	91.2
Path-heavy	(0.25, 0.15, 0.10, 0.35, 0.15)	88.5	90.9	92.1
Risk-low	(0.35, 0.20, 0.15, 0.20, 0.10)	87.6	88.5	91.4

Note. Sensitivity of recommendation metrics to different weight configurations (λ_1-λ_5).

The default setting is chosen because it provides the best overall Acc and Hit@3 while keeping CSR above 92%. Increasing λ_5 improves CSR slightly but reduces Acc by

over-penalizing candidates with incomplete attributes. Reducing λ_5 causes a clear CSR decrease, confirming the importance of risk control in food recommendation.

4.6. Efficiency Analysis with Module-Level Breakdown

Table 12

Average response time breakdown by module (seconds/query)

Method	Intent parsing	Retrieval	Constraint/ranking	LLM generation	Total ART
LLM-Direct	0.00	0.00	0.00	2.18	2.18
NR-KGNN	0.12	0.39	0.04	2.31	2.11
KG-CRAG (Ours)	0.14	0.48	0.17	1.94	2.73

Note. Detailed per-query execution time breakdown across modules. ART = Average Response Time.

This paper conducts a module-level efficiency analysis of the high-performing NR-KGNN method described in Section 4.3 against the most basic LLM-Direct baseline; the results are shown in Table 12. Compared with the classic LLM-Direct baseline, the proposed KG-CRAG method adds an additional 0.17 seconds to the processing time due to the introduction of explicit constraint computation and ranking modules; however, the overall time remains within the range of interactive responsiveness. The analysis indicates that the LLM generation phase remains the primary source of computational overhead across all methods. In the KG-CRAG method, the combined runtime of the KG retrieval and constraint/ranking modules is 0.65 seconds, accounting for approximately 23.8% of the total ART. This overhead contributes to the system's performance gains, and further optimisation could be achieved through strategies such as subgraph caching and batch generation in future work.

4.7. Quantitative Case Study

Table 13

Quantitative scores for example queries

Query	Method	Recommended dish	Acc	CSR	Faithfulness
Fat-loss, high-protein, dairy-free lunch	LLM-Direct	Yogurt chicken breast salad	0	0.75	0.50
Fat-loss, high-protein, dairy-free lunch	KG-CRAG	Chicken breast quinoa salad	1	1.00	1.00
Peanut allergy, light vegetarian dinner	NR-KGNN	Vegetarian noodles with peanut sauce	0	0.67	0.50

Peanut allergy, light vegetarian dinner	KG-CRAG	Mushroom and tofu soup with greens	1	1.00	1.00

Note. Qualitative and quantitative comparison of recommendations for representative sample queries.

This section presents a quantitative case study comparing our proposed method, KG-CRAG, with the high-performing NR-KGNN and the most basic baseline, LLM-Direct; the results are shown in Table 13. For the query regarding a low-fat, high-protein, dairy-free lunch, KG-CRAG accurately identified 'dairy-free' as a hard constraint, excluded options containing dairy ingredients such as yoghurt, and recommended a compliant chicken breast and quinoa salad, achieving Acc=1, CSR=1.00 and Faithfulness=1.00. In contrast, LLM-Direct failed to recognise this constraint and consequently provided an incorrect recommendation.For the peanut-allergy case, KG-CRAG identifies peanut allergy as a hard constraint, removes all dishes connected to peanut or nut allergen paths, and obtains CSR=1.00. The generated response explicitly states that peanuts and related nut ingredients have been excluded, which improves both safety and faithfulness.

4.8. Error Analysis

Table 14

Error categories for the remaining 21 KG-CRAG top-1 errors

Error source	Count	Proportion	Example	Future improvement
KG coverage gap	7	33.3%	Missing sauce-level allergen label	Expand ingredient and condiment relations
Intent/constraint parsing error	5	23.8%	Ambiguous phrase "not too heavy" parsed only as low fat	Add clarification dialogue and more parsing examples
Ranking trade-off error	4	19.0%	High protein chosen over taste preference	Learn weights from user feedback
LLM generation hallucination	3	14.3%	Generated unsupported cooking advice	Stricter evidence-only decoding template
Ambiguous or underspecified query	2	9.5%	No calorie target for "healthy dinner"	Ask follow-up questions

Note. Error breakdown of the 21 top-1 prediction errors made by KG-CRAG. KG = Knowledge Graph; LLM = Large Language Model.

The error analysis shows that most residual failures are caused by KG incompleteness and parsing ambiguity rather than the LLM alone. This finding suggests that improving KG coverage and interactive constraint clarification will be more effective than increasing model size only.

4.9. Discussion and Limitations

The experimental results show that the key challenge in food recommendation is not simply generating natural language answers but completing constraint identification, candidate filtering, and evidence retrieval before generation. The KG provides structured constraint-checking capability, while RAG converts retrieved evidence into user-readable explanations. Combining the two retains the interaction ability of LLMs and enhances controllability.

Several limitations remain. First, KG completeness is limited by the coverage of recipe descriptions and ingredient-level nutrition mapping. Second, Acc and Faithfulness involve subjective human evaluation, although the annotation guideline and Fleiss' Kappa results reduce this risk. Third, the current ranking weights are fixed; although sensitivity analysis is provided, future work should learn personalized weights from user feedback. Fourth, the present system does not yet fully integrate energy consumption and water footprint into the ranking function. If the work remains in a FEW Nexus venue, future versions should link the KG with cooking-energy coefficients and ingredient water-footprint datasets, so that recommendations can optimize nutrition, energy, and water sustainability jointly.

5. Conclusion

To address the issues of knowledge hallucination, insufficient constraint satisfaction, and weak interpretability of large language models in food recommendation, this paper proposes a personalized food recommendation method based on constraint-aware knowledge graph RAG. This method converts user natural language into structured constraints through requirement parsing, utilizes the knowledge graph to retrieve candidate dishes and evidence subgraphs, and designs a recommendation ranking function that integrates constraint matching, nutritional balance, user preferences, graph paths, and risk penalties. Finally, it generates interpretable meal suggestions via the RAG approach. Experimental results show that our method outperforms direct generation, standard vector RAG, and traditional KG-RAG methods in terms of recommendation accuracy, constraint satisfaction rate, and answer faithfulness. The method described in this paper also offers significant advantages over baseline approaches such as rule-based nutritional scoring, prompt-engineered LLMs, fine-tuned dense retrieval RAG, KGAT, FKGM and NR-KGNN. In the future, we plan to expand the dataset to include real-world user queries, incorporate more authoritative nutritional databases and expert review mechanisms, integrate cooking energy consumption and water footprint attributes into the knowledge graph, and learn personalised weights based

on user feedback. This will enhance the system's reliability in practical health management and food-energy-water resource coordination recommendation scenarios.

Acknowledgments

The authors have no additional acknowledgments to declare.

Author contributions

Conceptualization, Jia Yu and Tong Ji; methodology, Jia Yu; software, Jia Yu and Tong Ji; validation, Jia Yu, Tong Ji and Jiamin Zheng; formal analysis, Jia Yu; investigation, Jia Yu and Tong Ji; resources, Jiamin Zheng; data curation, Jia Yu and Tong Ji; writing—original draft preparation, Jia Yu; writing—review and editing, Jia Yu, Tong Ji and Jiamin Zheng; visualization, Jia Yu and Tong Ji; supervision, Jia Yu; project administration, Jia Yu; funding acquisition, Jia Yu. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the 2026 Undergraduate Innovation and Entrepreneurship Training Program of Beijing Technology and Business University, grant number S202610011005.

Data availability

No datasets were generated or analysed during the current study.

Declarations

Conflict of interest

The authors declare that there is no conflict of interest.

Ethics approval

Not applicable.

Consent to participate

Not applicable.

Consent for publication

All authors have given their consent.

Additional information

Received: 25 May 2026

Accepted: 18 June 2026

Food-Energy-Water Nexus
WISDOM ACADEMIC https://doi.org/10.63808/fewn.v2i3.428 Volume 2 , Issue 3
Published online: 16 August 2026

Publisher's Note

Wisdom Academic Press Ltd. remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[1] Ceccaldi, E., Niewiadomski, R., Mancini, M., & Volpe, G. (2022). What's on your plate? Collecting multimodal data to understand commensal behavior. *Frontiers in Psychology, 13*, Article 911000. https://doi.org/10.3389/fpsyg.2022.911000

[2] Misra, N. N., Dixit, Y., Al-Mallahi, A., Bhullar, M. S., Upadhyay, R., & Martynenko, A. (2020). IoT, big data, and artificial intelligence in agriculture and the food industry. *IEEE Internet of Things Journal, 9*(9), 6305–6324. https://doi.org/10.1109/JIOT.2020.2998584

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 5998–6008). https://doi.org/10.48550/arXiv.1706.03762

[4] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 1877–1901). https://doi.org/10.5555/3495724.3495883

[5] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *Proceedings of the 10th International Conference on Learning Representations*. https://openreview.net/forum?id=nZeVKeeFYf9

[6] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized large language models [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2305.14314

[7] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 9459–9474). https://doi.org/10.5555/3495724.3496517

[8] Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Van Den Driessche, G., Lespiau, J.-B., Damoc, B., Clark, A., de Las Casas, D., Guy, A., Menick, J., Ring, R., Hennigan, T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., … Sifre, L. (2022). Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning* (Vol. 162, pp. 2206–2240). PMLR. https://proceedings.mlr.press/v162/borgeaud22a.html

[9] Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., & Schulman, J. (2021). WebGPT: Browser-assisted question-answering with human feedback [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2112.09332

[10] Wang, Z., Teo, S. X., Chew, J. J., & Lim, Y. (2025). InstructRAG: Leveraging retrieval-augmented generation on instruction graphs for LLM-based task planning. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1413–1422). Association for Computing Machinery. https://arxiv.org/abs/2504.13032

[11] Min, W. Q., Liu, C. L., Xu, L. Y., & Jiang, S. Q. (2022). Applications of knowledge graphs for food science and industry. *Patterns, 3*(5), Article 100484. https://doi.org/10.1016/j.patter.2022.100484

[12] Ma, P., Tsai, S., He, Y., Jia, X., Zhen, D., Yu, N., Wang, Q., Ahuja, J. K. C., & Wei, C.-I. (2024). Large language models in food science: Innovations, applications, and future. *Trends in Food Science & Technology, 148*, Article 104488. https://doi.org/10.1016/j.tifs.2024.104488

[13] Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2024). Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering, 36*(7), 3580–3599. https://doi.org/10.1109/TKDE.2024.3352100

[14] Yang, W., Some, L., Bain, M., & Wang, Y. (2025). A comprehensive survey on integrating large language models with knowledge-based methods. *Knowledge-Based Systems*, Article 113503. https://doi.org/10.1016/j.knosys.2025.113503

[15] Peng, C. Y., Xia, F., Naseriparsa, M., & Osborne, F. (2023). Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review, 56*(11), 13071–13102. https://doi.org/10.1007/s10462-023-10465-9

[16] Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., Kirrane, S., Labra Gayo, J. E., Navigli, R., Neumaier, S., Ngonga Ngomo, A.-C., Polleres, A., Roomp, K., Sack, H., Sequeda, J., & Simperl, E. (2021). Knowledge graphs. *ACM Computing Surveys, 54*(4), 1–37. https://doi.org/10.1145/3447772

[17] Bao, J., Duan, N., Yan, Z., & Zhao, W. (2016). Constraint-based question answering with knowledge graph. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)* (pp. 2503–2514). Association for Computational Linguistics.

[18] Wang, X., He, X., Cao, Y., Liu, M., & Chua, T.-S. (2019). KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 950–958). Association for Computing Machinery. https://doi.org/10.1145/3292500.3330989

[19] Chen, Y., Guo, Y., Fan, Q., Zhang, Q., & Dong, Y. (2023). Health-aware food recommendation based on knowledge graph and multi-task learning. *Foods, 12*(10), Article 2079. https://doi.org/10.3390/foods12102079

[20] Ma, W., Li, M., Dai, J., Ding, J., Chu, Z., & Chen, H. (2024). Nutrition-related knowledge graph neural network for food recommendation. *Foods, 13*(13), Article 2144. https://doi.org/10.3390/foods13132144

[21] Felfernig, A., Friedrich, G., Jannach, D., & Zanker, M. (2015). Constraint-based recommender systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (2nd ed., pp. 161–197). Springer. https://doi.org/10.1007/978-1-4899-7637-6_5

[22] Hoekstra, A. Y., & Chapagain, A. K. (2008). *Globalization of water: Sharing the planet's freshwater resources*. Blackwell Publishing.

[23] U.S. Department of Agriculture, Agricultural Research Service, Beltsville Human Nutrition Research Center. (n.d.). *FoodData Central*. https://fdc.nal.usda.gov/