# Intelligent Recommendation and Process Generation Technology for Smart Restaurant Scenarios Based on Knowledge Graph

**Zhanqiang Liu[1], Liang Dong[2], Jianling Zhao[1,*], Jianguo Wang[3]**
[1]*Shandong Hehe Information Technology Co., Ltd., Jinan, China*
[2]*Jinan BOSAI Network Technology Co., Ltd., Jinan, China*
[3]*Shandong Tongyun Network Technology Co., Ltd., Jinan, China*
**Corresponding Author*

**Abstract: In an effort to refine dish recommendation personalization and automate service procedures in smart restaurants, a graph system is engineered. This system consolidates knowledge from diverse sources, including dishes, raw materials, customer preferences, and operational sequences. The study then examines the dish recommendation approach and the process generation mechanism via graph reasoning.This paper investigates the joint recommendation strategy via graph embedding and path confidence, and incorporates a directed operation graph for the automated generation and dissemination of task instructions.The empirical evidence indicates that the method excels in recommendation accuracy, response delay, and process completion rate, along with commendable semantic interpretability and system stability.The study finds that the method exhibits superior performance in recommendation accuracy, response delay, and process completion rate, along with commendable semantic interpretability and system stability.**

**Keywords: Smart Restaurant; Knowledge Graph; Dish Recommendation; Process Generation**

## 1. Introduction

The intelligent transformation of the food service industry necessitates deeper integration of personalized recommendation and automated service processes, especially in dynamic and heterogeneous dining scenarios. Traditional rule-based recommendation systems struggle to meet users' personalized demands and lack semantic understanding, making them less effective in supporting complex decision-making. Knowledge graphs (KGs), by offering structured semantic representations and reasoning capabilities, provide a robust theoretical foundation for building intelligent systems that can support both personalized recommendation and process automation. Recent studies have highlighted the effectiveness of KG-enhanced food recommendation systems. Lei et al. [1] proposed a multi-modal recipe recommendation framework that integrates user demand features with knowledge graphs, significantly improving relevance and diversity in recommendations. Miao et al. [2] further explored the integration of users' long-term and short-term interests into KG-based systems to enhance responsiveness in restaurant contexts. Meanwhile, Chen et al. [3] introduced a health-aware recommendation model combining KG and multi-task learning to ensure both personalization and nutritional balance. Motivated by these advancements, this study aims to construct a unified smart restaurant knowledge graph incorporating dish data, user preferences, ingredients, and service workflows. It further proposes a joint strategy for dish recommendation and process generation based on KG reasoning, targeting semantic interpretability and real-time adaptability.

## 2. Smart Restaurant Knowledge Graph Construction

### 2.1 Domain Analysis and Pattern Design

The construction of a smart restaurant knowledge graph begins with understanding the semantics of the food service domain and analyzing relationships among entities such as dishes, ingredients, customer preferences, and service processes. The ontology must balance expressive power and reasoning capability [1]. This study adopts a hierarchical ontology structure: the top level defines the ternary framework of "service object – behavior – state", while the lower level refines modules like "dishes – ingredients – processes", "user – preferences – dietary taboos", and "environment – time – dining state" (Figure 1).The dish module uses multi-dimensional attributes including taste (policyLevel), nutritionScore, and cookingTime. An attribute association formula is introduced:

$$R_{flavor}(c_i, a_j) = \frac{f(a_j) \cdot w_{ij}}{\sum_{k=1}^{n} f(a_k) \cdot w_{ik}} \quad (1)$$

Where ci denotes the ith dish, aj is its jth ingredient, f(aj) is the ingredient flavor intensity, and wij is the weight of the ingredient in the dish, reflecting the contribution to the style of the dish in the recommendation system. The schema is implemented using Neo4j's attribute graph model, enabling semantic queries and path reasoning. Relationships between modules are standardized through a relationship matrix (Table 1), supporting knowledge extraction, fusion, and seamless integration with the recommendation mechanism.

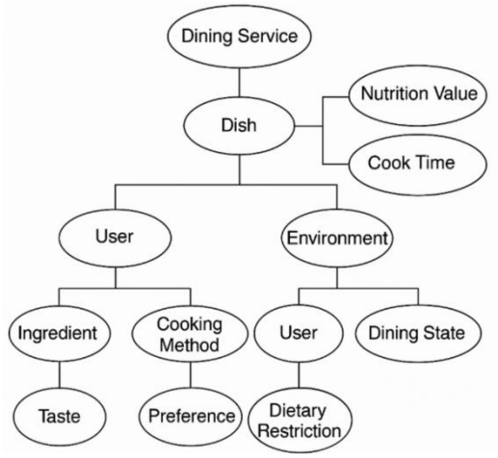


**Figure 1. Structure of Knowledge Graph Ontology Design for Smart Restaurant**

**Table 1. Knowledge Mapping Core Entity Relationship Mapping Matrix**

| Entity 1 | Entity 2 | Relationship Type |
|---|---|---|
| Dishes | Raw material | contains |
| Users | Preferences | has_preference |
| Dishes | Craftsmanship | cooked_by |
| Dining time | User status | occurs_in |
| Dish | Calorie | has_nutrition |

### 2.2 Multi-Source Data Acquisition and Knowledge Extraction

Following domain modeling, the smart restaurant knowledge graph relies on efficient data acquisition and extraction mechanisms to dynamically generate and update entity relationships. Data sources include structured data (e.g., recipe databases, nutrition tables), semi-structured data (e.g., dish APIs, ordering system logs), and unstructured data (e.g., user reviews, recipe images). These are integrated into a unified knowledge framework through multi-strategy fusion [2]. To ensure accuracy and scalability, a joint extraction model combining Named Entity Recognition (NER) and Dependency Syntax Analysis (DSA) is employed. Extraction rules are formally defined as:

$$T(e, r, e') = \{(e_i, r_{ij}, e_j) | e_i \in E, e_j \in E, r_{ij} \in R\} \quad (2)$$

where E is the set of extracted entities, R is the set of relations, and the ternary T denotes the semantic structural units parsed from the original corpus. For multimodal support, dish labels and image feature vectors are matched using the ResNet50 model to aid entity recognition.

### 2.3 Knowledge Fusion and Storage

After completing the multi-source data extraction, a knowledge fusion mechanism with high consistency and low redundancy needs to be designed to solve the problems of entity heteronymity, relationship conflict and contextual concept merging. To this end, the entity alignment strategy based on the semantic similarity of attribute vectors is adopted, and the entity fusion function is designed as follows:

$$Sim(e_i, e_j) = \frac{\sum_{k=1}^{n} w_k \cdot sim_k(a_i^k, a_j^k)}{\sum_{k=1}^{n} w_k} \quad (3)$$

Where ei,ej are candidate entities, aik,ajk are their kth attribute values, simkis the similarity function under the attribute, and wk is the attribute importance weight. Entity pairs are considered as the same object for merging when Sim> τ (fusion threshold) [3]. After structural unification, relational conflicts are disambiguated by a semantic rule base, and a collection of triples conforming to the Schema constraints of the graph is uniformly generated. The fused knowledge is stored in a graph database Neo4j, which adopts an attribute graph model to manage entity nodes, relationship edges and their attributes, and supports a hybrid query of Gremlin and Cypher. Figure 2 shows the processing flow of entity alignment, conflict resolution and ternary generalization in the fusion pipeline, which effectively supports the subsequent dynamic update mechanism and the graph-driven recommendation generation engine.

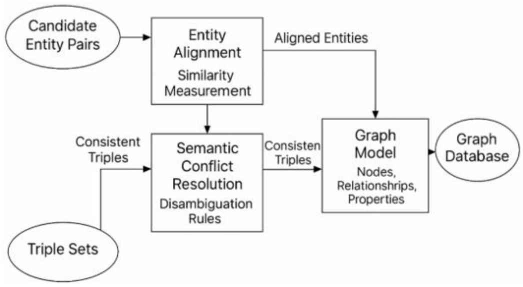


**Figure 2. Knowledge Fusion and Storage Flowchart**

## 3. Dish Intelligent Recommendation and Process Generation Technology

### 3.1 Overall System Architecture

The integrated architecture of dish intelligent recommendation and process generation for intelligent restaurant scenarios is organized in a modular asynchronous coupling manner, which mainly includes five core components such as user image modeling module, map query and inference engine, recommendation strategy scheduler, process generation unit and feedback iteration controller [4]. The system adopts event-driven architecture (EDA) to listen to user behavior in real time, semantically maps the structured state of the dining scenario to the middle layer of the graph, and drives the personalized dish recommendation through the combination of Path Reasoning and Graph Embedding mechanism. The generated results are then combined with the kitchen process specification by the process builder to generate the task instruction chain.

### 3.2 Knowledge Graph-Based Dish Recommendation

The dish recommendation module builds an inference link based on the structured semantics of the graph and the context of user behavior to achieve accurate personalized recommendation service. The recommendation task is formalized as a graph query problem under path constraints, i.e., in the knowledge graph G=(E,R), based on the behavioral and pictorial features of user nodes u ∈ E, search for a set of related dish nodes Du⊆E under the premise of satisfying the constraints C. In this study, we design a joint recommendation strategy incorporating graph embedding and semantic paths, and the recommendation scoring function is defined as follows:

$$Score(u,d) = \alpha \cdot Sim(\vec{u},\vec{d}) + \beta \cdot P_{KG}(u,d) \tag{4}$$

Where, $\vec{u}$ ,$\vec{d}$ are the graph embedding vectors of users and dishes respectively, Sim denotes the vector similarity (cosine similarity is used), P(KG) (u,d) denotes the path confidence score from users to dishes in the graph, and α , β are the adjustable weight parameters. The path confidence is jointly computed by combining the path length, relationship type weights and the importance of entities in the graph to guarantee the semantic interpretability of the recommendation [5]. Figure 3 shows the multi-hop inference structure that starts from the user node and reaches the candidate dishes through the "preference-ingredient-taste" path, which is performed by the path generator calling the knowledge graph inference engine in real time to perform the query operation, and outputs the candidate set to be passed to the recommendation strategy scheduler. The module design reserves collaborative filtering and reinforcement learning recommendation interfaces to enhance the system's ability to adapt to complex dietary scenarios and provide highly semantically related input parameters for service flow generation.

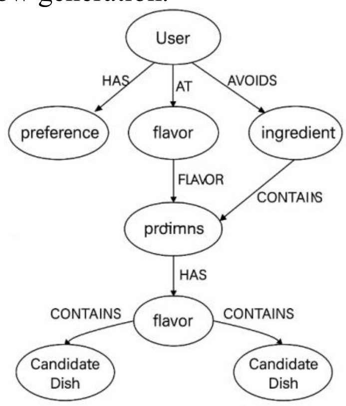


**Figure 3. Structure of User-Driven Path Reasoning for Dish Recommendation**

### 3.3 Process Generation Technology Based on Recommendation Results

Based on the set Du={d1,d2,.......,dn} of dish recommendation results output in Section 2.2, the process generation module needs to automatically construct an execution process containing operation nodes, dependency constraints, and resource scheduling commands based on the graphical structure of each dish and the service rules of the back office. To this end, this study introduces a process generation strategy that integrates rule ontology and workflow graph modeling, where each dish is mapped as a Directed Operation Graph (DOG), with nodes denoting processing actions (e.g., preprocessing, cooking, and plating), and edges denoting time/resource dependencies between tasks. The process scheduling function is defined as follows:

$$T_{task}(d_i) = \arg\min_{\pi \in \Pi_i} \left( \sum_{j=1}^{|\pi|} \tau_j^{exec} + \sum_{(j,k)\in E_\pi} \delta_{jk}^{wait} \right) \quad (5)$$

Where Πi denotes all possible operation paths of dish di, τ jexec is the operation execution delay of the jth step, and δ jkwait is the waiting delay between operations. The system adopts the "dish-raw material-process-equipment" relationship in the knowledge graph to construct the process skeleton, and matches the adapted templates through the rule engine, and finally outputs them in the form of structured task instructions. The process is released through the message bus as a chain of kitchen commands, realizing automated collaboration with the back office terminals, and at the same time, the execution receipt is fed back to the map to complete the closed-loop update.

## 4. Experimental Validation and Effect Evaluation

In order to verify the effectiveness of the recommendation and process generation system driven by the knowledge graph of the constructed smart restaurant, this paper builds an experimental platform based on real dining behavior logs and meal process libraries, and adopts a modular distributed deployment method to reproduce the whole process of "behavioral acquisition - graph reasoning - recommendation output - process generation - feedback scheduling". The whole process of "behavior collection - map inference - recommendation output - process generation - feedback scheduling" is reproduced. The experimental scenarios cover the two peak dining periods of lunch and dinner on weekdays. Taking the user click logs and service execution records as benchmarks, the system performance is comprehensively evaluated by multi-dimensional indicators, including the recommendation accuracy (Precision@K), the dish click conversion rate (CTR), the average process scheduling latency (Avg Latency), and the Task Completion Rate. Rate), etc. As shown in Table 2, the experimental platform evaluates the recommendation effect by comparing the graph-enhanced recommendation algorithm (KG-RS) with the traditional collaborative filtering model (CF-Baseline), and evaluates the stability of the process scheduling module by taking the success rate of process generation and the process latency as the key indicators. Figure 4 further visualizes the fluctuation trend of click response efficiency of different recommendation strategies under different time periods.

**Table 2. Comparison of Recommendation and Process Scheduling Module Effectiveness Indicators**

| Model/Indicator | Precision@5 | CTR (%) | Avg Latency (ms) | Task Completion Rate (%) |
|---|---|---|---|---|
| KG-RS (Graph Driven Recommendation) | 0.842 | 38.5 | 173 | 96.2 |
| CF-Baseline | 0.711 | 31.8 | 239 | 89.5 |

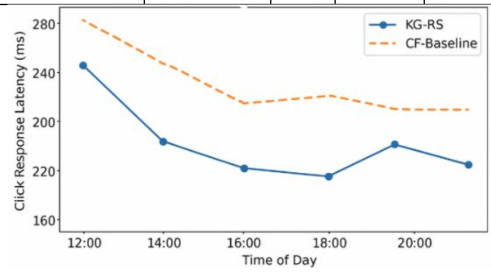


**Figure 4. Trend of Click Response Latency for Different Strategies with Time Period Distribution**

## 5. Conclusion

The construction of a knowledge graph-centered smart restaurant system effectively integrates semantic reasoning with automated process generation, achieving significant improvements in recommendation accuracy and service efficiency. The current framework demonstrates robust performance in typical scenarios, as validated by high task completion rates and low latency. However, the system's adaptability in extreme high-frequency or large-scale update environments remains to be further enhanced. Future work will focus on optimizing real-time graph update mechanisms and incorporating reinforcement-based reasoning to improve robustness and scalability in more dynamic and multimodal contexts.

## References

[1]Sha, F., Ding, C., Zheng, X., Wang, J., & Tao, Y. (2025). Weathering the Policy Storm: How Trade Uncertainty Shapes Firm Financial Performance through Innovation and Operations. International Review of Economics & Finance, 104274.

[2]Sha, F., Meng, J., Zheng, X., & Jiang, Y.

(2025). Sustainability Under Fire: How China-US Tensions Impact Corporate ESG Performance?. Finance Research Letters, 107882.

[3]Zhang, L., & Meng, Q. (2025). User Portrait-Driven Smart Home Device Deployment Optimization and Spatial Interaction Design.