



# 本科生毕业设计（论文）

论文题目 基于知识图谱的饮食推荐系统  
的设计与实现

作者姓名 王晓哲  
专业 计算机科学与技术  
指导教师 陈子军教授

2024 年 6 月

# 燕山大学本科生毕业设计（论文）

## 基于知识图谱的饮食推荐系统的设计与实现

学 院：信息科学与工程学院  
专 业：计算机科学与技术  
姓 名：王晓哲  
学 号：202011040034  
指 导 教 师：陈子军教授  
答 辩 日 期：2024 年 6 月

## 学位论文原创性声明

郑重声明：所呈交的学位论文《基于知识图谱的饮食推荐系统的设计与实现》，是本人在导师的指导下，独立进行研究取得的成果。除文中已经注明引用的内容外，本论文不包括他人或集体已经发表或撰写过的作品成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果，并承诺因本声明而产生的法律结果由本人承担。

学位论文作者签名：王晓哲

日期：2024年5月30日

## 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权燕山大学将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保 密□，在\_\_年解密后适用本授权书。

本学位论文属于

不保密

(请在以上相应方框内打“√”)

学位论文作者签名：王晓哲

日期：2024年5月30日

指导教师签名：王晓哲

日期：2024年5月30日

## 燕山大学毕业设计(论文)任务书

学院：信息科学与工程学院

专业(方向): 计算机科学与技术

学号	202011040034	学生姓名	王晓哲	专业班级	20 级计算机 2 班		
题目	题目名称	基于知识图谱的饮食推荐系统的设计与实现					
	题目性质	1.理工类：工程设计（ ）；工程技术实验研究型（ ）；理论研究型（ ）；计算机软件型（√）；综合型（ ）。 2.文管类（ ）；3.外语类（ ）；4.艺术类（ ）。					
	题目类型	1.毕业设计（√）		2.论文（ ）			
	题目来源	科研课题（ ）		生产实际（ ）	自选题目（√）		
主要内容	设计并实现基于知识图谱的饮食推荐系统，通过利用知识图谱技术，根据用户的个人健康状态、饮食偏好以及营养需求，提出个性化、科学合理的饮食建议。主要功能有： 1. 用户和食谱信息收集：收集用户的基本信息、身体状况、饮食偏好等数据。 2. 知识图谱构建：整合食材、工艺等维度信息，构建饮食知识图谱。 3. 食谱推荐：基于用户的喜好类别和点赞收藏食谱列表进行食谱推荐。 4. 食谱浏览：用户通过系统浏览食谱，点赞和收藏食谱，浏览食谱的详细信息。 5. 社区互动与分享：建立用户社区，让用户分享饮食心得、厨艺经验，并进行互动。						
基本要求	1. 系统使用方便简单且适合不同层次、不同使用习惯的用户使用，界面简洁美观。 2. 系统在结构上应该具有很好的可扩展性，以便于以后的进一步开发和维护。 3. 软件设计符合法律、行业规范，充分考虑用户需求。 4. 在系统设计过程中，进行方案设计与对比分析，充分考虑开发成本和维护成本。 5. 论文内容编写和格式符合燕山大学本科论文说明书规则。						
参考资料	[1] 龙中华. Spring Boot 实战派. 电子工业出版社, 2020. [2] 孙卫琴. Tomcat 与 Java Web 开发技术详解（第 3 版）. 电子工业出版社. 2019. [3] 王昊奋,漆桂林,陈华钧. 知识图谱：方法、实践与应用. 电子工业出版社. 2019. [4] 陈红,王珊,张孝. 数据库系统原理教程（第 2 版）. 清华大学出版社. 2021.						
周次	12月12日—寒假	1—3周	4—9周	10—13周	14—15周		
应完成的内容	1.确定选题 2.选题主要内容 3.选题基本要求 4.选题设计思路	1.需求分析 2.模块设计 3.数据集准备 4.数据库设计 5.撰写报告 6.准备开题答辩	1.详细设计 2.程序接口设计 3.模型训练和评价 4.编码和单元测试 5.撰写中期报告 6.准备中期答辩	1.系统完善 2.详细测试 3.整理文档 4.撰写毕业设计论文	1.整理论文和相关资料 2.毕业答辩		

## 摘要

在中国经济的高速增长势头下，人民对优质饮食生活的追求不断升级。个性化和多元化的饮食推荐、分享与交流平台，正是满足这一需求的关键工具。此平台不仅能够激发饮食消费潜力，还将为推动国内饮食市场的发展提供有力支撑。基于知识图谱的饮食推荐系统旨在构建一个食谱获取和分享的饮食平台，为用户提供一个交流氛围友好，用户体验良好的饮食交流社区。

本系统采用 Java、SpringBoot、Vue.js、Flask 等主流开发技术和框架，利用 MySQL 数据库和 Redis 缓存数据库，实现系统的基本功能，满足系统后续的升级和扩展。本项目采用 Scrapy 框架，深入分析“美食杰”网站组成，构建本系统食谱数据集和用户-偏好数据集，利用 RippleNet 模型和共有属性采样数据增强算法，重构用户-食谱交互记录，训练食谱推荐模型。系统还利用 Neo4j 提供的图算法，挖掘节点间的相似度权重，反应用户潜在的食谱喜好，为用户提供全方位的饮食推荐结果。

本系统分为用户端和后台管理员端。用户可以通过系统进行点赞、收藏食谱，和参与到社区交流中去，并获得个性化推荐得食谱。管理员通过后台系统保证系统数据的安全性和准确性。本项目旨在营造一个和谐的社区交流氛围，增强用户间的联系，增强网站的文化饮食底蕴，满足人们对美好饮食的需要。

**关键词：**知识图谱； RippleNet 模型； 食谱分享平台； Vue 框架； Spring Boot 框架；  
前后端分离开发

## Abstract

With the rapid growth momentum of China's economy, people's pursuit of high-quality diet life is constantly upgrading. Personalized and diversified diet recommendation, sharing and communication platform is a key tool to meet this demand. This platform can not only stimulate the potential of diet consumption, but also provide strong support for the development of the domestic diet market. The diet recommendation system based on knowledge graph aims to build a diet platform for recipe acquisition and sharing, and provide users with a diet communication community with friendly communication atmosphere and good user experience.

The system uses Java, SpringBoot, Vue.js, Flask and other mainstream development technologies and frameworks, and uses MySQL database and Redis cache database to realize the basic functions of the system and meet the subsequent upgrade and expansion of the system. This project uses the Scrapy framework, deeply analyzes the composition of the "Gourmet" website, constructs the recipe data set and user-preference data set of this system, and uses the RippleNet model and the common attribute sampling data enhancement algorithm to reconstruct the user-recipe interaction records and train the recipe recommendation model. The system also uses the graph algorithm provided by Neo4j to mine the similarity weights between nodes, reflect the user's potential recipe preferences, and provide users with comprehensive diet recommendation results.

This system is divided into user end and background administrator end. Users can use the system to like, collect recipes, participate in community communication, and get personalized recommended recipes. Administrators ensure the security and accuracy of system data through the background system, create a harmonious community communication atmosphere, enhance the contact between users, and enhance the cultural diet of the website

**Keywords:** Knowledge graph; RippleNet model; Recipe sharing platform; Vue framework;  
SpringBoot framework; Front-end and back-end separate

## 目 录

摘 要 .....	I
Abstract.....	II
第 1 章 绪 论 .....	1
1.1 课题的背景 .....	1
1.2 目的和意义 .....	2
1.3 国内外文献综述 .....	2
1.4 主要研究内容 .....	3
1.4.1 研究目标 .....	3
1.4.2 研究内容 .....	3
第 2 章 关键技术概述 .....	5
2.1 前后端分离开发模式 .....	5
2.2 相关技术介绍 .....	5
2.2.1 Spring Boot 框架 .....	5
2.2.2 Vue 开发框架 .....	6
2.2.3 MySQL 数据库 .....	7
2.2.4 Redis 数据库 .....	7
2.2.5 MyBatis 框架 .....	7
2.3 本课题方案设计及对比 .....	9
2.4 本章小结 .....	10
第 3 章 系统需求分析 .....	11
3.1 系统总体需求 .....	11
3.2 用户分析 .....	12
3.3 系统功能需求分析 .....	13
3.3.1 普通用户功能性需求分析 .....	13
3.3.2 管理员功能性需求分析 .....	15
3.4 系统非功能性需求分析 .....	16
3.5 本章小结 .....	17
第 4 章 系统的总体设计 .....	19
4.1 系统逻辑架构设计 .....	19
4.2 系统技术架构设计 .....	20
4.3 系统功能模块设计 .....	21

---

4.4 数据库设计 .....	22
4.4.1 系统 E-R 图设计 .....	22
4.4.2 数据库表设计 .....	23
4.5 本章小结 .....	28
第 5 章 基于知识图谱的饮食推荐 .....	29
5.1 RippleNet.....	29
5.1.1 RippleNet 介绍.....	29
5.1.2 用户兴趣建模 .....	30
5.1.3 共有属性采样算法 .....	32
5.2 数据集的获取与处理 .....	32
5.2.1 数据集的获取 .....	32
5.2.2 数据集预处理 .....	35
5.2.3 构建知识图谱 .....	35
5.2.4 数据集增强 .....	37
5.3 实验与分析 .....	38
5.3.1 实验环境 .....	38
5.3.2 实验结果及分析 .....	38
5.4 基于 Neo4j 个性化饮食推荐 .....	39
5.5 本章小结 .....	42
第 6 章 系统的详细设计与实现 .....	43
6.1 开发环境搭建 .....	43
6.1.1 前端环境搭建 .....	43
6.1.2 后端环境搭建 .....	44
6.1.3 数据库搭建和配置 .....	46
6.2 登录注册详细设计与实现 .....	47
6.2.1 登录注册的流程设计 .....	47
6.2.2 登录注册的类设计和实现 .....	48
6.3 用户端系统详细设计与实现 .....	51
6.3.1 食谱浏览模块 .....	51
6.3.2 社区交流模块 .....	56
6.3.3 个人中心模块 .....	59
6.4 后台管理系统详细设计与实现 .....	63
6.4.1 用户管理模块 .....	63
6.4.2 权限管理模块 .....	65
6.4.3 食谱管理模块 .....	68
6.4.4 社区交流模块 .....	69

## 目 录

---

6.5 饮食推荐模块 .....	69
6.5.1 为新用户进行推荐 .....	69
6.5.2 为老用户进行推荐 .....	70
6.6 成功展示 .....	71
6.7 本章小结 .....	75
<b>第 7 章 系统测试 .....</b>	<b>77</b>
7.1 系统测试环境说明 .....	77
7.2 系统功能测试用户 .....	77
7.2.1 用户端功能测试 .....	77
7.2.2 管理员端功能测试 .....	80
7.3 本章小结 .....	81
<b>结 论 .....</b>	<b>83</b>
<b>参考文献 .....</b>	<b>85</b>
<b>致 谢 .....</b>	<b>87</b>
<b>附录 1 开题报告 .....</b>	<b>89</b>
<b>附录 2 文献综述 .....</b>	<b>99</b>
<b>附录 3 中期报告 .....</b>	<b>103</b>
<b>附录 4 外文原文 .....</b>	<b>127</b>
<b>附录 5 外文翻译 .....</b>	<b>135</b>



## 第1章 绪 论

### 1.1 课题的背景

党的十八大以来，随着中国的改革开放不断深入推进，居民收入水平较快增长，人民收入结构不断改善，人民生活质量大幅度提高，在人民日常生活中，民以食为天，饮食扮演着至关重要的角色，饮食健康逐渐成为人民普遍关注的热点问题，对饮食的个性化和多样化的需求也不断提高。中国特色社会主义社会进入新时代，我国的主要矛盾已经转化为人民日益增长的美好生活需要和不平衡不充分的发展之间的矛盾。健康饮食已成为日益增长的公众关注焦点，越来越多的人群开始关注个人饮食健康和食品健康。

新中国成立以来，尤其是改革开放的推行，推动了我国的健康事业的飞速发展，居民的主要健康指标普遍超越了中高收入国家的平均水平。尽管如此，国民在健康知识的普及程度仍然不足，普遍存在饮食不均衡和缺少体育锻炼等不健康的生活习惯。在 2015 年 10 月，党的十八届五中全会上，明确了加快健康中国建设的战略目标，这一决策是在“五位一体”的总体布局和“四个全面”战略布局的基础上作出的，能够更有效保障人民健康状况。随着“健康中国”战略的不断推进实施，中国营养健康食品行业呈现出前所未有的蓬勃发展态势。《中国营养健康食品行业蓝皮书》（以下简称蓝皮书）揭示了健康食品行业的规模、发展态势和挑战，同时也展现了当前健康饮食发展的巨大潜力。蓝皮书指出，中国营养健康食品行业规模已突破五千亿元大关，在未来几年，该行业将持续稳步增长。这一发展趋势，离不开消费者对健康生活的追求以及国家各方面政策的支持等方面因素。中国居民饮食习惯普遍存在多糖、多盐、多油问题，易引发糖尿病、高血压、肥胖等健康问题，国家将健康膳食作为战略目标，出台相关政策和规划明确发展营养健康行业<sup>[1]</sup>。

同时，随着互联网和手机移动技术的发展，人们也可以随时随地获取丰富的食谱（也可成为菜谱）的相关信息，而这样海量食谱信息也难以满足用户的个性化需求，如果从这些海量的信息中获取到对个人有价值的食谱内容，已经成为一个亟待解决的难题。传统的搜索引擎只是通过关键词与内容的相关度和匹配度，来帮助用户筛选

和过滤信息，在一定程度上，能够降低用户获取饮食信息的能力，但没有提供个性化的服务，仍需要用户去手动筛选符合个人的食谱。

## 1.2 目的和意义

相比于传统的搜索引擎，饮食推荐系统不仅能够解决食谱信息的超载问题，还能满足人民对饮食的个性化、多元化的多层次需求。同时，使用饮食知识图谱可以为系统提供有效处理传统推荐系统存在的食谱数据稀疏、冷启动、推荐结构可解释性差等问题。食谱知识图谱包含了食谱名称、主料、辅料、难度、工艺等与食谱有关的信息，系统通过用户的喜好类别和交互列表为用户推荐符合用户口味的潜在兴趣食谱，从而满足用户个性化的饮食需求，提高用户的使用体验，激发用户的探索欲。

在当前存在的饮食推荐系统下，且使用饮食知识图谱能有效提高传统推荐算法的准确率的背景下，本项目通过爬虫获取数据，构建本系统的饮食知识图谱和用户与食谱的偏好数据集，使用知识图谱相关的推荐算法和模型，设计并实现一个基于知识图谱的饮食推荐系统，为用户提供日常个性化、多样化、符合用户喜好的食谱。

## 1.3 国内外文献综述

在国外饮食方面的研究中，相关食谱的推荐出现较早。1995 年，Epicurious 食谱网站初步建成获得大量的好评。随后，Food、Allrecipes 等一众食谱网站接踵而至，引起网络食谱热潮，从此时起，很多关于饮食推荐系统的研究相继问世<sup>[2]</sup>。在 2010 年，美国麻省理工学院计算机科学和人工智能实验室的研究人员和哈佛大学公共卫生学院的营养专家一同发布了 Recipe1M+<sup>[3]</sup>公开数据集，此数据集有一百万份食谱数据和一千三百万份大规模的食品图像数据集。文献[4]使用 Recipe1M+ 数据集进行了基于内容的推荐算法和协同过滤推荐算法的研究，实现并设计了获取相似用户的偏好食谱的功能模型。文献[5]利用 Recipe1M+ 数据集利用相关技术构建了饮食知识图谱，并开源了相关知识图谱的源程序和数据，为其他研究者提供了饮食知识图谱的理论依据和数据支持。文献[6]通过爬取 Allrecipe、Food 等网站的食谱数据，构建了 MealRec 饮食数据集，并提出了一种基于类别约束的食谱推荐模型。MealRec 饮食数据集包含了五百多位用户和七千多份食谱，该研究为后续他人的研究提供理论基础。

国内的食谱分布广泛，随地域有所差异，种类较多，信息较杂，关于食谱的专业性更强。国内有关饮食推荐系统的开展的研究还比较初步，相关的饮食数据较少。随

着中华美食网（2003 年建立）、豆果网（2008 年建立）等饮食食谱交流网站的蓬勃发展，国内的饮食推荐领域的研究有了数据来源和支持<sup>[2]</sup>。文献[2]利用共有属性采样算法，加强了用户与食谱间的历史交互记录数据，改善了用户与食谱交互不充分的问题，RippleNet 模型的预测效果有了显著增强。文献[7]使用爬虫框架技术爬取美食杰网站的食谱信息数据，自行构建了食谱知识图谱，使用有关上下文和知识表示的联合学习方法进行模型训练，完成了模型的部署，实现基于知识图谱的个性化饮食推荐。国内的饮食推荐领域相较于国外的推荐领域数据集广泛，食谱多样化，种类繁多，工艺较杂，研究范围广，专业性强，更具有研究价值。

## 1.4 主要研究内容

### 1.4.1 研究目标

为了满足人民对美好生活中的饮食获取和交流的需求，并利用饮食知识图谱在提高个性化推荐效果上的优点，研究并实现基于知识图谱的饮食推荐系统。在推荐的基础上，根据用户的操作行为表现，为用户提供食谱推荐服务。为了增强用户的交互性和娱乐性，系统增加社区交流模块，可以分享日常饮食攻略、饮食注意事项等内容。

### 1.4.2 研究内容

基于上述研究目标，本文的研究内容和方向主要有以下八个方面：

- (1) 通过爬取美食杰网站信息，利用半结构化数据，构建提前设定的关系的知识图谱，建立本系统的食谱数据集。
- (2) 构建用户-饮食偏好数据集，并对知识图谱数据进行预处理。通过爬取美食杰等网站获取数据，创建用户的交互数据，即用户-食谱偏好数据集。
- (3) 寻找合适的数据存储形式。饮食知识图谱可以使用图数据库进行存储，如 Neo4j；用户-饮食偏好数据集可以使用 txt 文档进行存储。
- (4) 以构建的数据集为基础，实现结合知识图谱的推荐模型 RippleNet，完成用户的个性化饮食推荐结果生成，并能定期更新数据，继续训练模型并使用。
- (5) 设计和实现 B/S 架构的 Web 系统，包括用户登录、注册、查询和浏览饮食信息等基本功能。用户在系统中能够点赞和收藏食谱，并可以标记食谱喜好类别。
- (6) 结合上述 Web 系统，完成推荐模型的调用，为用户提供个性化推荐服务。

- (7) 设计和实现社区交流模块，使用户可以畅所欲言，分享日常饮食、注意事项等内容。用户可以进行发布主题、回复主题等基本操作。
- (8) 设计和实现管理员后台系统，实现管理用户、饮食信息、社区交流等内容，维护网站的信息安全，营造和谐的分享和交流氛围。

## 第 2 章 关键技术概述

本章将对基于知识图谱的饮食推荐系统开发过程中涉及的关键技术进行阐述，其中主要包括：前后端分离开发模式、SpringBoot 框架、Vue 框架、MyBatis 框架、MySQL 数据库、Redis 数据库，并描述以上技术在项目中的实现原理及其特点。同时，在本章结尾，将本项目前期拟定的两种实现方案从技术难度、技术先进程度、开发和维护成本等方面进行对比，最终选择前后端分离开发方案展开实施。

### 2.1 前后端分离开发模式

在传统的 Java Web 开发模式中，前端代码（HTML, JavaScript, CSS）通常与后端代码紧密耦合在一起，这种开发模式在一定程度上限制了开发的灵活性和扩展性。在这种模式下，当需要对交互功能进行修改或优化时，可能会影响整个系统的功能，造成牵一发而动全身的问题。

随着 Java web 开发技术的不断进步和发展，前后端分离开发成为开发人员最为青睐的开发模式之一。在这种开发模式下，前端项目可以利用后端提供的接口实现前端页面和功能的开发，而后端项目则只需要关注后端业务逻辑的处理。这种开发模式大大提高了开发效率和代码质量，为后续项目的扩展和维护奠定基础。前后端分离开发模式也会使代码更加模块化和清晰，有助于后期维护，降低了代码的耦合度。

### 2.2 相关技术介绍

#### 2.2.1 Spring Boot 框架

本系统的后端部分使用 Spring Boot 框架完成功能逻辑开发工作。SpringBoot 框架是基于 Spring 框架二次开发的框架，它简化了 Spring 应用的创建、配置的流程，可以一站式解决 Web 后端的开发，提供了较为齐全的 Spring 模块支持。它对 Spring 框架的主体功能完全保留，继承了 Spring 框架的两大特性：IOC（Inversion of Control）和 AOP（Aspect Oriented Programming）。IOC，名为控制反转，它将由程序控制对象的创建和调用的权力赋予给容器，容器接管了对象的生命周期，完成对象组件的配

置和管理，降低了代码的耦合度。AOP 即面向切面编程，它是通过预编译方式和运行期动态 diamagnetic 来实现程序功能的统一维护的一种技术。AOP 技术可以使得业务逻辑各个部分相互隔离，使得各部分之间耦合度降低，提高程序的可扩展性、可读性以及灵活性，使开发人员能够更高效、更快捷完成相应编码工作。

此外，SpringBoot 还支持多种框架和插件，内嵌了 Tomcat 服务器自动进行 Web 服务部署和相关配置工作。Swagger 是一个 RESTful 接口文档的规范和工作集，能够方便开发人员查看和使用接口的功能，帮助前后端开发协同工作，提高开发效率。在 SpringBoot 应用中，项目可以集成 Swagger 插件，再通过向接口和相关类中添加注解自动完成生成接口文档的工作，极大提高了开发效率。SpringBoot 应用可以使用 SpringSecurity 完成用户的认证和授权，并通过 YML 文件统一管理各个框架参数。在基于框架的 Javaweb 项目中不可避免地要使用大量的第三方 jar 包，需要使用 Maven 进行项目构建和依赖管理<sup>[9]</sup>。

因此，SpringBoot 框架旨在帮助开发人员快速完成开发，省去繁琐的配置工作流程，快速完成项目的搭建，是目前最为流行的开发框架之一。

## 2.2.2 Vue 开发框架

在传统的 HTML5 页面的开发过程中，业务逻辑内容和页面渲染无法同时进行，且用户长时间在同一界面容易出现长时间处于静止未响应状态，用户体验感不佳，而 Vue 开发框架致力于构建数据驱动的渐进式 Web 应用开发框架，可以自底向上逐层应用，只需关注视图层，开发人员易于上手，用户体验感较好。Vue 可以实现数据双向绑定，即当数据发生变化时，视图也随之发生变化；当视图发生变化时，数据也会随之改变。

Vue 框架是一款由国人开发的框架，相比于传统的 Web 开发方式，Vue 提供了更加简洁、高效的开发方式，凭借其先进的技术和出色的性能表现，在业界引起了广泛的关注和认可<sup>[10]</sup>。Vue 遵循 CMD 规范，也提供了 MVVM 设计模式（数据双向绑定），这使得 Vue 在开发过程中拥有更好的模块化管理特性，在大型项目开发过程中利于开发人员维护管理。在 MVVM 模式下，Vue 实现了数据和视图的分离，并通过数据绑定机制实现了双向绑定<sup>[10]</sup>。这不仅让开发人员更便捷地进行数据处理和页面设计工作，也提高了数据的渲染效率和页面性能表现。Vue 框架相较于前端框架，上手较快，支持多款插件和依赖库，可扩展性强。

### 2.2.3 MySQL 数据库

MySQL 是 Oracle 公司开发的一款开源的关系型数据库管理系统。MySQL 数据库具有速度较快、成本较低，体积较小的特点，它所具有的功能已能满足比较复杂的应用，因此，广泛应用于软件开发中<sup>[11]</sup>。MySQL 支持多个操作系统，包括 Windows、Linux、macOS 等，并且提供了高效、稳定的数据存储和管理服务，是目前市面最流行的开源关系型数据库之一，被广泛用于 Web 应用程序、企业应用程序等领域，受到业界广泛关注。

### 2.2.4 Redis 数据库

Redis 数据库是现在最受欢迎的 NoSQL 数据库之一。Redis 是以 key-value 方式存储数据的一种非关系型（NoSQL）数据库。它的数据是无结构的，通常直接在内存当中进行读写，因此查询速度非常快<sup>[11]</sup>。随着互联网技术和 5G 技术的不断成熟，每天都有海量的信息需要被访问，数据规模越来越庞大，传统的数据库在应对过多数据访问场景时容易出现访问时间长、性能大幅度下降等问题。Redis 以高速读写以及分布式存储的特性解决了这一缺陷，具有较高的查询效率，缓解了数据库的访问读取压力。当系统需要承受的访问数据库的并发量很大时，普通关系型数据库可能承受不了高强度的请求，从而导致关系型数据库访存压力过大，耗时较长，为减少大量数据的 I/O 请求的资源消耗，需要把数据从硬盘转移到内存当中<sup>[12]</sup>，以提高处理并发的能力。

Redis 数据库支持 5 种数据类型结构，能够根据不同场景的需要选择不同的数据类型，并实现高效的存储效果，弥补了其他内存型数据库存储类型方面的不足。特别的，Redis 支持两种持久化模式，分别为 RDB 模式和 AOF 模式，可以完成多个服务器对主服务器的数据库中数据进行拷贝。此外，Redis 的执行命令过程中都会按照顺序进行，不会被其他服务器发送的命令请求所打断，这在一定程度上保证了数据库数据的一致性和稳定性。

### 2.2.5 MyBatis 框架

MyBatis 是一个操作数据库的持久层框架，该框架支持自定义 SQL 语句，自定义存储过程以及完成高级映射的配置，简化了项目的开发流程。传统的 Java 数据库连接（Java Database Connectivity，简称 JDBC）需要手动对查询结果集进行封装及完

成数据库的连接和关闭操作，极大增加了编程的重复工作量，使得操作数据库的过程过于繁琐。MyBatis 对 JDBC 进行了二次封装，能够与第三方数据库连接池集成，实现 Connection 对象的复用<sup>[13]</sup>。MyBatis 使开发人员将主要精力放在编写 SQL 语句上，通过映射方式，半自动化地生成持久化代码<sup>[14]</sup>。

Mybatis 是一个持久层的 ORM 框架，它是在 JDBC 的基础上，进行了二次封装和开发。它采用 DAO 模式将从 MySQL 数据库中查询到的字段与 Java 中的实体类建立映射（ORM 映射），完成对象数据和关系数据的转化，是一种半自动化的 ORM 映射。同时，MyBatis 支持多种标签，在编写 SQL 语句过程中，可以动态生成 SQL 语句，无需重复编写繁琐的依据条件查询、删除等 SQL 语句，只需要通过标签判断条件即可生成动态 SQL，完成对数据库的访存操作。Mybatis 更为容易地被广大开发人员接受，关键就在于 Mybatis 操作简单，在引入 Spring Boot 框架之后，开发人员只需通过编辑配置文件信息即可完成项目与 Mybatis 框架的整合工作<sup>[15]</sup>。

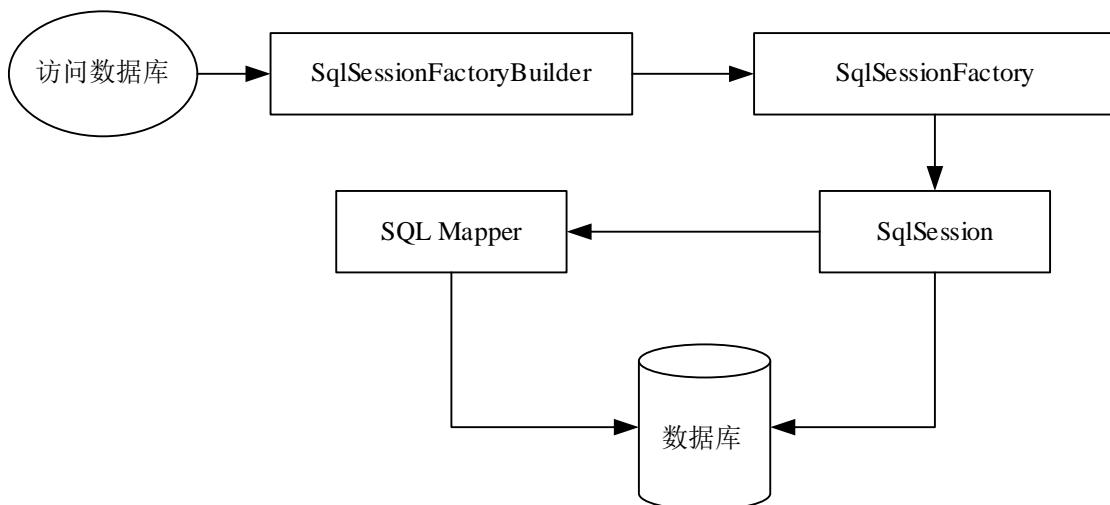


图 2-1 MyBatis 核心原理图

如图 2-1 所示，为 MyBatis 的核心原理图。首先，MyBatis 框架会优先读取处于资源包下的 yml 配置文件或 properties 配置文件，读取其中有关连接数据库的 URL，并与 MySQL 建立连接。随后，SqlSessionFactoryBuilder 会创建出 SqlSessionFactory，然后，再由 SqlSessionFactory 创建相关的 SqlSession。在 SqlSession 内部封装了关于 JDBC 操作的程序内容。在进行业务逻辑编写时，只需要使用 SqlSession 即可完成 CRUD 的操作。为减少资源的浪费和保证事务的一致性，SqlSession 会在对数据库操作后进行关闭，它仅存活于一个应用的请求和操作中。

## 2.3 本课题方案设计及对比

在本课题实施的过程中，拟定通过时间调研法，深入实际需求，获取一手资料，了解本课题实际情况和需求，总结课题过程中可以遇到的问题，并借助文献研究法，广泛查阅与本课题相关的文献资料，借鉴前人的研究成果和方法，拓宽研究视野，总结规划本课题的实施方案，不断总结经验教训，发现问题并解释解决，优化课题实施流程，提高实施效率。

本课题拟定实现一个 Web 形式的饮食推荐系统。目前，开发 Web 应用的技术多种多样，依据课题的调研结果和技术熟练程度设计了以下两种实施方案。

(1) 方案 1 使用 JSP 动态页面技术，将 Java 代码和动态渲染页面数据的程序嵌入到静态页面中，实现以 `jsp` 文件为模板，动态生成和渲染页面上展示的数据。JSP 页面既可以定义 HTML、JavaScript、CSS 等静态页面内容，也可以定义有关数据库操作的动态内容。

(2) 方案 2 使用 SpringBoot、SpringSecurity 等框架搭建后端服务器，完成用户认证、授权等基本工作，并负责与 MySQL 和 Redis 数据库进行交互，实现数据的持久化操作。使用 Vue.js、Node.js、Element-Plus 等技术搭建前端项目，实现一个交互性良好、界面美观的饮食推荐系统的网站。根据目前主流的 Web 开发框架 Spring MVC 分层设计的思想，本项目在系统的总体架构上可以初步分为视图层（View）、控制层（Controller）和模型层（Model）。视图层负责展示食谱数据、主贴数据等内容，并处理用户对页面的交互。控制层负责发送用户操作页面时发送的请求，从而调用模型层中与之对应的相关接口的服务，从而完成对数据库的访存业务逻辑处理，并将响应结果返回给视图层，最终由视图层渲染到页面中，展示给用户。模型层完成数据库的访存工作，并封装相关的业务逻辑功能。系统项目代码又可以分为交互层、业务逻辑层和数据层。其中，视图层与交互层相对应，模型层可以被划分为数据层和业务逻辑层，控制层工作在交互层和业务逻辑层之间。。

在方案 1 中，JSP 技术可以在网页中编写操作数据库的 Java 代码，并很轻松将其展现在网页中，但是 Java 代码耦合在静态页面中，耦合度较强，代码独立性较差，极不利于后续的维护和扩展。而在方案 2 中，前后端分离并行开发，互不干扰。在进行前端开发时，无需关注后端如何实现，只需要使用后端暴露的接口即可，反之亦然。通过方案 2 开发的项目，代码耦合度降低，代码结构更加清晰明了，易于后期的维护

和扩展。所以，本课题采用方案 2 的设计方案。

## 2.4 本章小结

本章介绍了基于知识图谱的饮食推荐系统所使用的相关技术，其中包括 SpringBoot 框架、Vue 框架、MyBatis 框架、MySQL 数据库、Redis 数据库等。最后，根据本项目的业务逻辑特点和 Web 特性，提出了两种实施方案，并通过分析两种实施方案的难易程度、可扩展性、稳定性等内容，最终选择了使用 SpringBoot 框架等技术来进行本项目的实施工作，以下章节都是围绕该方案进行展开实施。

## 第3章 系统需求分析

系统需求分析是指对一个系统进行详细的调查和研究，以确定开发的软件系统能够满足用户的需求和期望，是软件生命周期重要的阶段之一<sup>[17]</sup>。这一阶段需要将系统所面向用户的需求以计算机思维进行归纳整理。通过需求分析，明确系统的详细功能划分，梳理出完整的需求文档，确定需要实现的功能，为后续工作奠定基础。

本章将从需求层面，对基于知识图谱的饮食推荐系统作出全面的需求分析，并通过绘制用例图地方式进行功能性设计和分析，依照系统的实际需要，描述系统的非功能性需求分析。

### 3.1 系统总体需求

本系统的主要目标是为用户提供一个能够获取食谱信息，并可以基于用户的历 史点赞和收藏行为得到系统为用户个性化推荐的食谱信息的平台。同时，本系统也为 用户提供了一个社区交流平台，不同用户之间可以交流美食心得和饮食健康知识。后 台管理员可以对这些用户信息、食谱信息和社区交流信息进行统一管理。



图 3-1 系统功能划分图

如图 3-1 所示，为系统的总体功能划分。饮食推荐系统将划分为 7 大功能。

**(1) 数据集获取** 本系统需要构建饮食知识图谱和用户食谱交互偏好数据集，这两类数据都可以从各大美食平台进行爬取，一部分是用户交互记录，另一部分是饮食详细信息。这两类数据集在爬取完成后，需要进行数据的筛选和清理工作，去除空值、无意义值等无效信息。

**(2) 知识图谱创建** 根据提前设置的知识图谱的实体和关系结构，将获取的饮食详细信息提取相关的实体和关系，构造饮食知识图谱三元组文件，然后将其导入到图数据库 Neo4j 中，以便为后续的推荐工作提供数据支持。

**(3) 饮食推荐** 在该功能下，无论是新注册用户和老用户都可以获取到个性

化饮食推荐信息。对于新注册用户，只需要在首次登录时，标记喜好类别便可通过在知识图谱中找寻准确拟推荐信息。对于老用户，系统会分析老用户的历史点赞和收藏记录，获取用户较为偏爱的类别，然后进行食谱推荐工作。

**(4) 食谱管理** 用户能够查看其它用户发布的食谱信息，可以进行点赞或收藏操作。用户在自己的个人主页可以查看自己所发布、点赞和收藏的食谱信息。管理员可以对已发布且存在某些问题的食谱进行修改或删除操作，以维护系统信息的准确性，避免误导用户。

**(5) 社区管理** 用户可以在社区中畅所欲言，发表自己最近关于饮食的所思所想所感所悟，也可以查看其他用户发表的主题内容，也可以进行评论。管理员则可以在后台管理系统对用户发表的主题和回复的内容进行管理，可以删除有害信息，维护良好、健康、和谐的社区交流环境。

**(6) 用户信息管理** 对于用户而言，用户可以修改个人资料，修改登录密码，也可以更换个人头像。管理员可以通过后台系统对系统中不活跃用户进行删除，可以修改用户的个人资料和重置用户的密码。在此功能下，后台管理系统可以新增多个不同的角色，并给不同的角色分配不同的后台功能菜单，使不同的角色管理员分管不同的模块，各司其职，使得系统信息更加稳定和安全。

**(7) 用户注册和登录** 登录注册功能是系统的最基本的功能。未登录用户虽然可以查看食谱信息，但是点赞收藏和发布食谱的功能是不能使用的。所以，用户需要输入必要的个人信息来进行注册。管理员的账号经过超级管理员的授权便可登录到后台管理系统中，根据该管理员所属角色拥有的菜单来初始化界面。

## 3.2 用户分析

本项目主要分为前台用户端和后台管理端，前台端和后台管理端之间共享数据。前台端的使用者主要为上网获取食谱的用户；后台管理系统的使用者为管理本网站的管理人员。该管理人员可以分为不同的角色，每个角色分管不同的功能模块，以确保管理员能够精准管控网站的信息。如图 3-2 所示，为本系统的用户角色图。

前台用户端主要为想要获取食谱的用户提供食谱浏览、个性化食谱推荐、食谱信息发布、参与社区交流等功能。后台系统主要允许网站管理员登录使用，不同的管理者对应了不同的管理角色，各自分管不同的业务模块。本项目的后台管理系统主要有三种管理员：第一种管理员是系统管理员，拥有系统的最高权限，可以使用后台管理

系统所有的功能，并能为其他管理员分配不同的业务模块，主要负责对用户、角色和菜单进行管理；第二种管理员是食谱管理员，该管理员需要具备一定的美食鉴别和烹饪能力，能够对系统中存在错误、不健康的食谱信息进行筛选和更正，并及时上传最近的各大短视频平台的火爆食谱菜系，以定期更新网站信息；第三种管理员是社区管理员，该管理员具备一定的文学功底和文章鉴别力，负责区分和筛选社区交流中有害信息，并加以删除，撰写美食文稿，上传系统，营造积极向上的社区氛围。

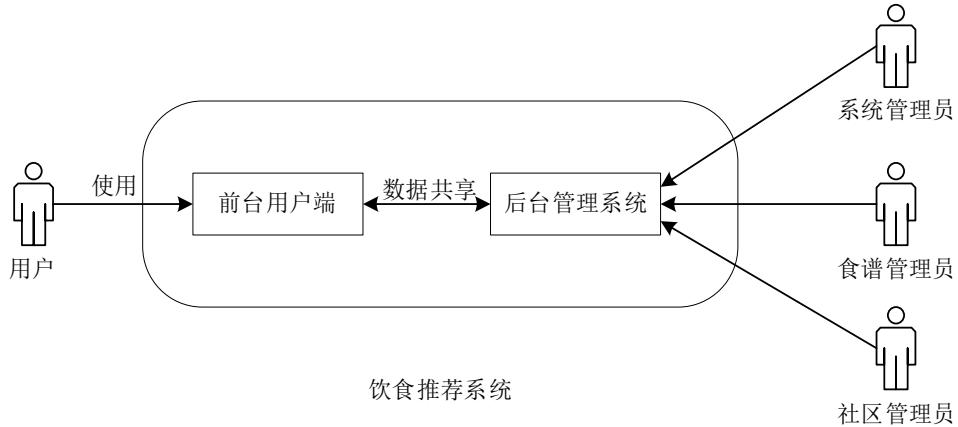


图 3-2 系统角色分布图

### 3.3 系统功能需求分析

上一节主要是对本系统所可能具有的角色和每种角色应该实现哪些功能进行简要介绍。为避免描述时功能之间的重叠，本节将上节提到三种管理员的功能进行合并为一位管理员进行阐述说明。本节将通过针对普通用户和管理员两种用户角色进行功能性需求分析，详细说明这两种角色的功能需求，并使用用例图进行详细解释和说明。

#### 3.3.1 普通用户功能性需求分析

本项目的用户端系统主要面向希望获取到食谱信息并参与饮食社区交流的用户使用，提供了一系列的食谱相关的服务。用户可以登录注册，浏览、点赞和收藏食谱，发布食谱，进行个人信息的修改，参与到社区交流，获取个性化食谱推荐。特别的，用户可以在系统中查看其它用户的点赞、收藏和发布的食谱信息，并能够查看该用户在社区中所发布的内容，以加深用户之间的互动和深入交流。如图 3-3 所示，为普通用户对应的用例图。这一小节将以食谱管理、社区交流和饮食推荐三个主要功能为例对普通用户进行功能性需求分析，详细描述这三个主要功能所应包含的其他子功能。

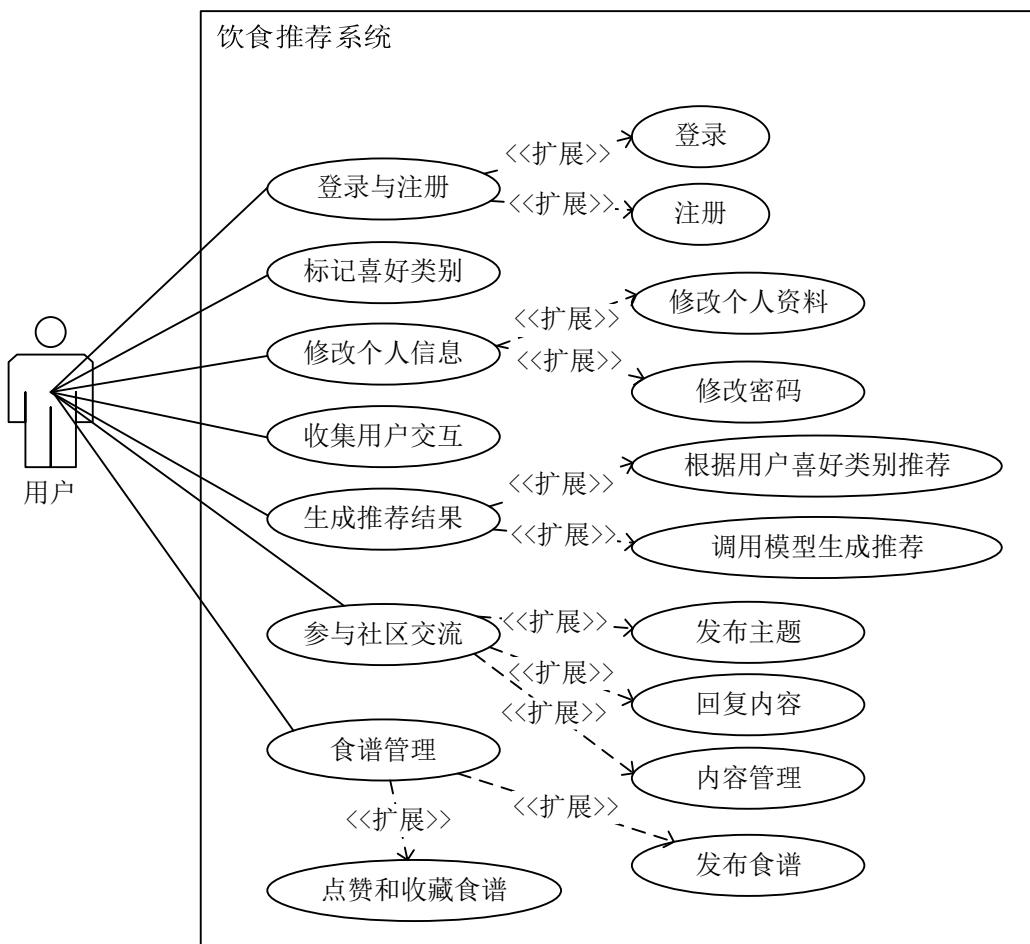


图 3-3 用户用例图

**(1) 食谱管理** 未登录用户和登录用户均可以对系统中存在的食谱信息进行浏览和查看，并可以通过食谱的一些分类，如工艺、味道、难度等级和所需时间进行检索，同样也可以根据食材的关键字检索出用户想要的食谱信息。这样可以在一定程度上帮助用户大致筛选出想要获取的食谱信息。

登录用户可以对自己喜爱的食谱进行点赞和收藏，以便后续能够快速找到自己已操作的食谱，同时也为系统对该用户进行食谱推荐提供数据支持。如果用户有自己所喜爱且想要分享的食谱，可以通过网站发布食谱的功能进行发布。

**(2) 社区交流** 大多数用户往往希望能够通过分享自己的所思所悟所感，获得其他网友的评价和共鸣，社区交流功能由此应运而生。用户可以在交流社区中发布自己的一些想法和平时对于饮食的见解，其他用户则可以对他人发布的信息进行回复。用户也可以对已经发布的社区内容和回复进行删除操作。这在很大程度上能够提高用户的归属感，让用户能够乐在其中，获得更多关于饮食上的看法。

(3) 饮食推荐 饮食推荐，顾名思义，即是为用户推荐其可能较为喜爱的食谱内容。对于新注册到系统的用户，用户需要根据自己平时的喜好，标注自己的个人喜好类别，以解决推荐中冷启动问题。系统则会根据新注册用户的个人喜好，通过推荐算法在知识图谱中检索到用户可能较为喜爱的食谱，进而向新注册用户进行展示。老用户由于较长时间使用了系统，点赞和收藏了一些食谱，系统会根据老用户的点赞和收藏列表解析出用户可能较为喜欢的喜好类别，系统根据喜好类别得到推荐食谱列表，并展示给用户。

### 3.3.2 管理员功能性需求分析

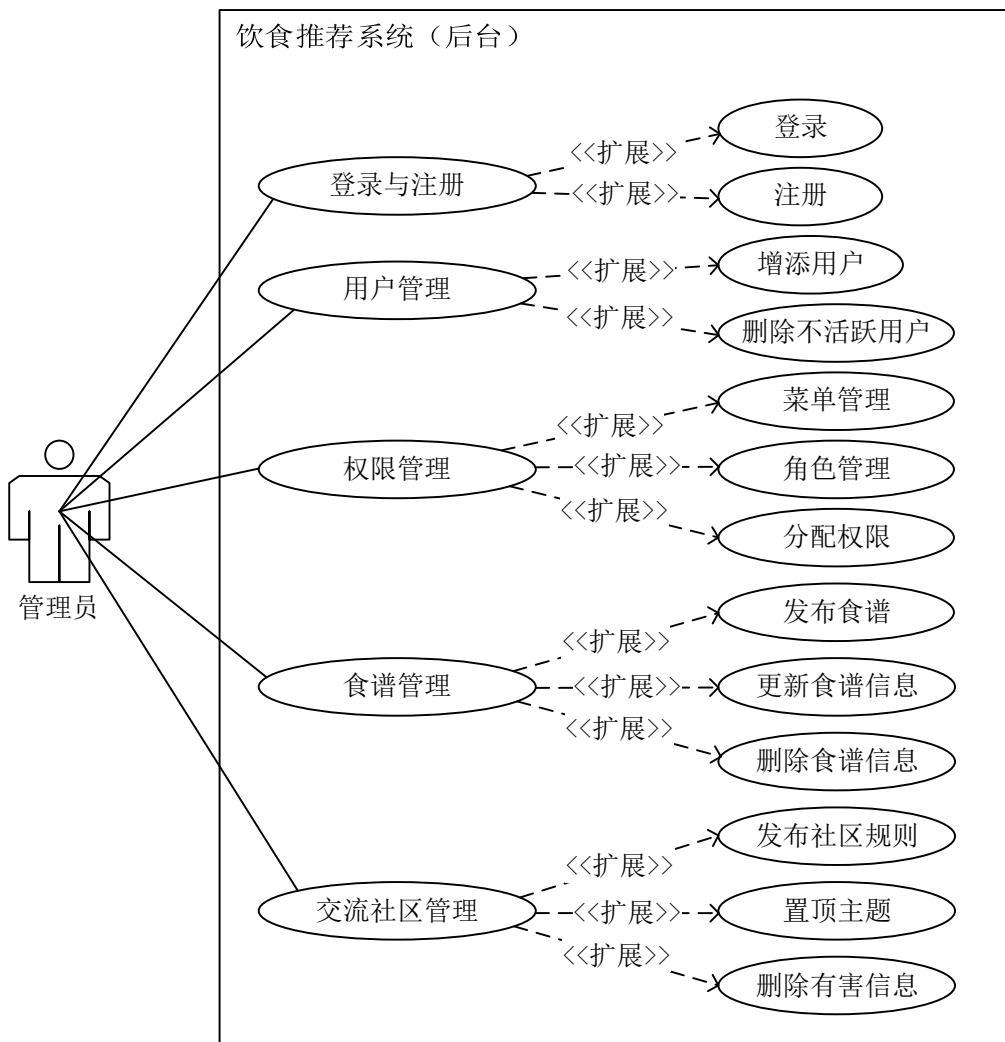


图 3-4 管理员用例图

本系统的管理员拟定有三种管理员，分别为系统管理员、食谱管理员和社区管理员。如图 3-4 所示的管理员用例图中，系统管理员一般有系统后台的最高权限，能够

对用户、菜单、角色和分配权限进行操作，而食谱管理员和社区管理员则分管食谱管理模块和交流社区模块。这三种管理员各司其职，职责明确，利于系统的信息安全和稳定。本节将对用户管理、权限管理、食谱管理和交流社区管理进行功能性需求分析。

**(1) 用户管理** 用户管理模块主要完成对注册到系统中的用户进行信息管理操作。在该模块中，管理员能够查看用户的展示列表，并可以通过用户名、性别、地址等信息检索用户。其次，该模块通过对最后登录时间作为筛选条件，寻找长时间未登录的用户，将其加以删除，释放系统存储空间。在必要时，管理员可以进行修改用户的个人信息和重置密码的操作。

**(2) 权限管理** 在该模块中，具有此模块管理权限的管理员可以根据不同的角色需要赋予不同的菜单权限，也可以新增或删除角色，来满足实际的需要，能够完成对角色信息进行修改更新。

**(3) 食谱管理** 管理员可以在食谱管理模块中新增最新的食谱内容，包括食谱的图片、名称、主料、辅料、工艺、味道等信息。当用户自行发布的某些食谱有不当或错误内容时，管理员可以进行修改或删除操作，以确保系统信息的正确性和安全性。此外，管理员也可以通过一些主料、辅料、名称的关键字或具体的工艺、味道等字段信息检索食谱信息，进而对这些食谱信息进行操作。

**(4) 交流社区管理** 在该模块中，管理员可以对用户所发布的主题信息进行筛选和修改，修改一些敏感词汇以确保社区交流环境的稳定。管理员还可以发布社区交流规则。

### 3.4 系统非功能性需求分析

系统除了要满足业务的功能需求之外，还需要保证能够满足具体的非功能需求，其中包括系统的安全性、可靠性、交互性、稳定性和健壮性等。系统的非功能需求往往影响着软件产品是否能够高效准确地运行，并向目标用户提供更加稳定的服务。因此，非功能性需求分析是需求分析中重要的组成部分。本节将从系统的简洁性、并发性、稳定性等几个方面对本系统的非功能性需求进行分析。

**(1) 系统并发性** 系统数据库在某一个时间段内被大量的用户同时访问的处理能力往往能够体现一个系统是否能够应对并发性的处理。本系统是一个开放性食谱网站，可能在某一时间段出现大规模访问，这就需要系统有足够的应对并发的能力。本系统拟引入 Redis 数据库以减轻那些会被高频访问的数据库表的压力，让系统在有

效时间内对用户发送的请求作出响应。系统拟引入 SpringBoot 的事务管理，保证数据的一致性和可靠性。

(2) **界面简洁性** 复杂的交互往往会使用户产生疲劳感，并对网站的操作流程不熟悉、不清晰。减少交互在一定程度上能够增强用户对系统的感兴趣程度。因此，在设计页面时应该减少交互，让界面尽量简洁，使用户能够在短时间内就能够了解系统的操作流程。

(3) **安全性** 系统的数据库表记录了用户的姓名、电话、地址、登录密码等重要的个人信息。在处理用户数据时，应当注意数据的安全性。例如，在进行密码操作时，应当对密码进行加密，避免个人信息的泄露，保障系统的信息安全。

(4) **准确性** 系统包含了大量的食谱信息，应当确保这些数据的准确性，以免错误的信息误导用户，避免产生不必要的误会。同时，管理员也应该把控交流社区环境绿色化、积极化，营造正能量的、积极向上的社区交流氛围。

### 3.5 本章小结

本章主要对饮食推荐系统进行了功能性需求分析和非功能性需求分析。首先具体详细描述了本系统包含的角色和每种角色具备的功能，并从业务功能逻辑出发详细描述了每种角色具备的主要功能。最后，从系统的并发性、界面的简洁性、系统的安全性和准确性四个维度出发，对系统进行了非功能性需求分析，为后续详细的软件设计开发工作奠定基础。



## 第4章 系统的总体设计

本章以上一章的需求分析结果为基础进行软件系统的总体设计。总体设计的主要任务是将需求分析的功能描述和用例图转化为对应的系统架构设计，并划分系统的基本功能模块。本章将依据上一章节的需求分析结果从系统的逻辑架构、系统的技术架构、系统功能结构和数据库设计四个方面进行总体设计。

### 4.1 系统逻辑架构设计

如图 4-1 所示，为系统的逻辑架构设计图。本项目的 Web 前端显示和后台管理端在系统逻辑结构上由上到下可以分为用户层、表现层、控制层、业务层和数据层。

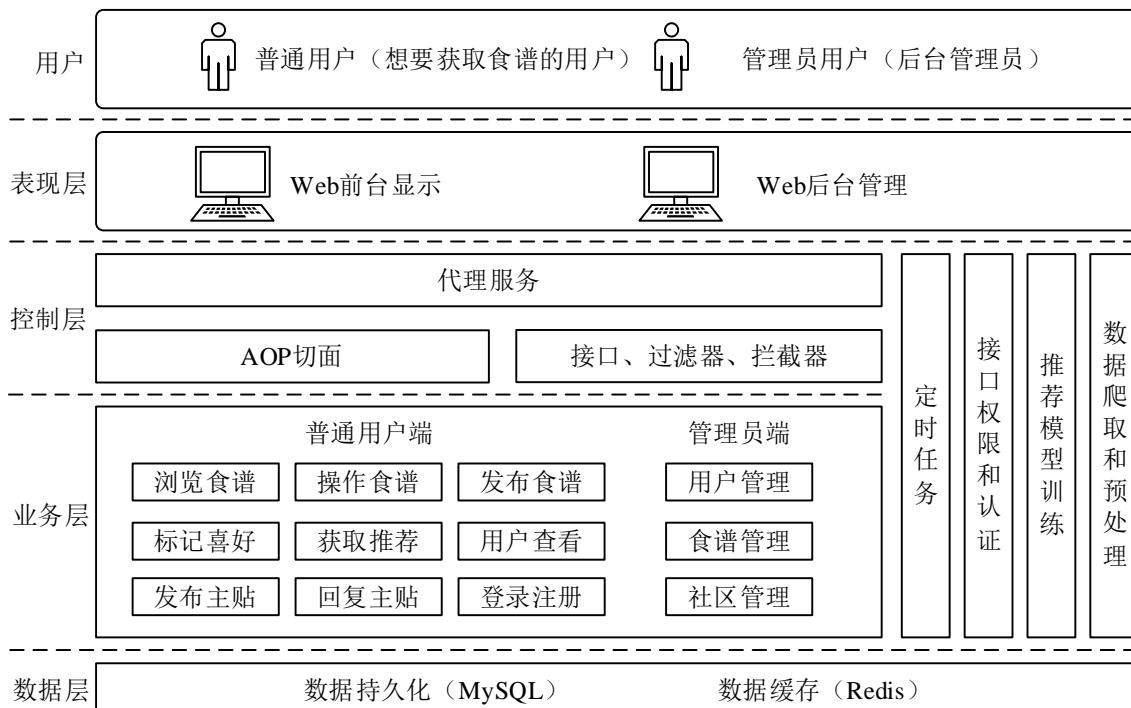


图 4-1 系统逻辑架构图

其中，表现层主要是通过页面等直观方式向用户和管理员提供接入系统的操作入口，包括有食谱需求的普通用户所使用的用户端系统和管理员的后台管理系统。控制层主要用于对表现层中的所发送的请求进行处理，并将处理结果反馈至表现层展示给用户。在前后端开发设计模式下，由于浏览器的同源策略，前后端必须要进行配置代理服务才可以进行通信，并且需要通过接口权限和认证模块对不同用户赋予不

同接口的权限，以保证系统的安全可靠，同时需要设置定时任务以确保数据缓存中的数据能够及时写入数据库完成持久化操作。在系统之外，本项目还需完成食谱数据的爬取和预处理工作，并通过处理后的食谱数据训练推荐模型。为了弥补面向对象编程的不足，本系统利用了 AOP（面向切面编程）对某些接口进行功能增强，降低了业务逻辑各部分之间的耦合度，提高了程序的可重用性。

业务层主要负责业务逻辑的处理，由于 Spring 框架具有 IOC 的特点，不仅能够更好地实现低耦合的特性，同时可以使代码更加简洁<sup>[18]</sup>。本系统的业务逻辑层分为普通用户端和管理员端，普通用户端主要包括浏览食谱、点赞和收藏食谱、发布食谱、标记喜好、获取推荐，查看其它用户主页等模块，该业务层负责对用户端发送的请求进行业务逻辑操作，从而与数据库实现交互。管理员端主要分为用户管理、食谱管理和社区管理。数据层主要负责对系统所产生和展示的数据进行存储，并通过引入 Redis 数据库缓解系统并发时数据库的压力，提高系统的安全性、并发性和稳定性。

## 4.2 系统技术架构设计

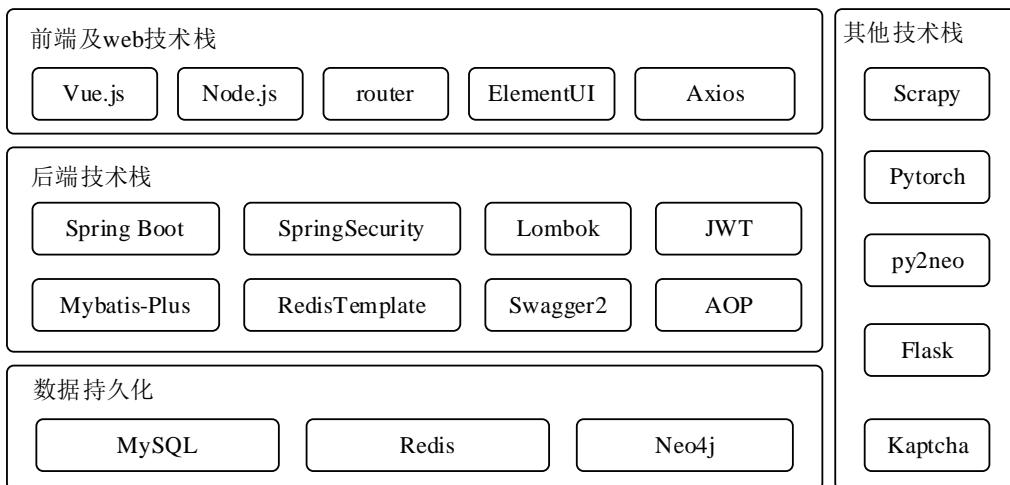


图 4-2 系统技术栈

如图 4-2 所示为系统技术栈。普通用户端和管理员端的展示层主要采用图 4-2 所示的前端及 web 技术栈。前端主要使用 Vue.js、Node.js 和 ElementUI 实现，使用 router 完成前端路由的动态配置。Axios 主要为前端提供发送请求的工具。后端使用 SpringBoot 技术框架进行搭建，并集成了 MyBatis 框架用于建立与 MySQL 数据库的连接以完成数据持久化操作。后端还集成了 SpringSecurity 安全框架用于配置系统的用户权限认证。此外，后端集成了其它组件来实现系统的其它业务功能需求。特别的，为了方便后端业务逻辑程序的构建和测试，本系统在后端项目中集成了 Swagger 插

件，自动生成 Restful 风格的在线文档 Web 服务。PageHelper 插件用来完成 SQL 语句的分页查询和数据封装。为了提高系统的并发性和系统性能，本系统引入 Redis 缓存数据库用于缓解 MySQL 数据库的缓存压力。RedisTemplate 是 Java 用来操作 Redis 的高度封装类，提供了对 key 绑定的便捷式操作 API。

### 4.3 系统功能模块设计

依据上文的需求分析和功能描述，本项目的主要功能模块图主要分为用户端和管理员端两部分内容。如图 4-3 所示为用户端系统功能模块图，其中包含食谱浏览、社区交流和个人中心三大功能模块。在食谱浏览中，登录用户可以获取到当前系统网站热门的食谱，也可以通过个人标记的喜好类别或用户的历史点赞和收藏记录获取系统为用户推荐的食谱信息。用户可以在浏览中查看某一食谱的具体信息，如发布人、食谱主食材、食谱辅食材、适宜人群等内容，并可以对某一食谱进行点赞和收藏操作。如果用户想要获取具体的食谱，如获取关于土豆的食谱，可以通过搜索框进行关键词搜索，或通过食谱的类别进行搜索。在社区交流中，用户可以查看其他用户发布的主贴，并查看主贴下具体的回复内容。用户可以在交流社区中，分类别发布自己的所思所感和求助饮食健康问题。在个人中心模块中，用户可以修改个人资料和修改登录密码，并且可以标注自己的喜好类别以获取食谱推荐结果。在此模块中，用户不仅可以发布和删除自己已发布的食谱信息，并且可以查看自己点赞和收藏的食谱数据。特别的，用户可以进入其他用户的主页，查看该用户点赞、收藏和发布的食谱。

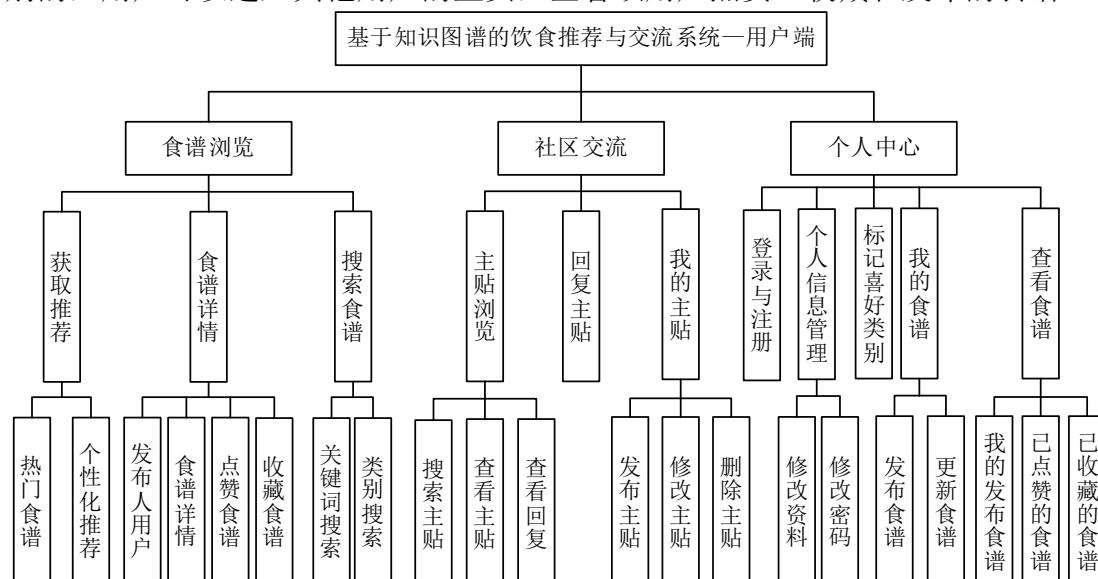


图 4-3 用户端系统功能模块图

如图 4-4 为管理员端系统功能模块图。食谱获取和饮食推荐两个模块内容是在开发项目时进行开展完成的，并未集成在后台管理系统中。在本项目中，管理员需要爬取相关的食谱数据并用程序将其进行预处理后导入 MySQL 数据库中，此外还需要对推荐模型进行训练，同时也可以依据基于入度排序进行推荐，基于文本相似度进行推荐，最终将推荐的业务逻辑进行部署，方便用户端进行访问和使用。

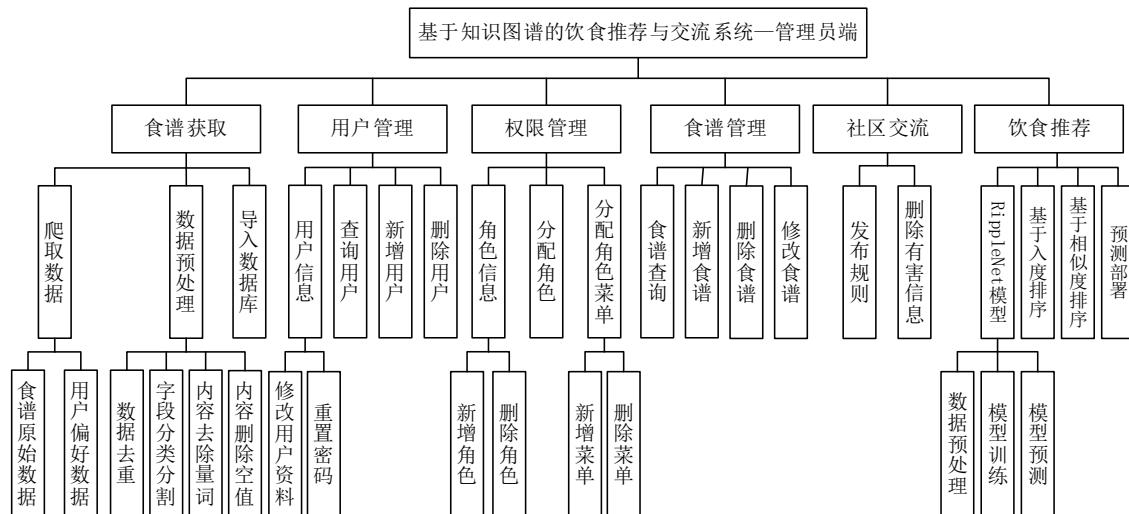


图 4-4 管理员端系统功能模块图

后台管理系统主要包含用户管理、权限管理、食谱管理和社区交流管理四大功能模块。在用户管理中，管理员可以对用户信息进行修改，并且可以删除长时间不活跃的用户。权限管理模块主要用于控制各个不同角色的管理员分管的菜单权限。在此模块下，管理员可以为角色分配不同的菜单从而管理不同的功能。在食谱管理模块中，管理员可以对食谱的信息进行修改，能够删除出现错误信息的食谱数据。管理员可以在社区交流中删除有害的信息，并能够发布相关规则，约束用户的行为，营造积极向上的交流氛围。

## 4.4 数据库设计

### 4.4.1 系统 E-R 图设计

本项目使用 MySQL 作为数据库用于数据的持久化操作，使用 Navicat 软件作为数据库管理和调试的工具。本节将从实体关系图（E-R 图）和数据库表设计两个方面来阐述本项目的数据库设计。首先，依照上述的需求分析和功能模块图设计，抽象出项目中可能包含的实体和其包含的属性信息，并建立实体之间的关联关系，绘制出 E-R 图，完成数据库的概念结构设计内容。其次，根据绘制的 E-R 图和每个属性的字

段特征构建具体的数据库表，完成数据库逻辑结构设计内容。

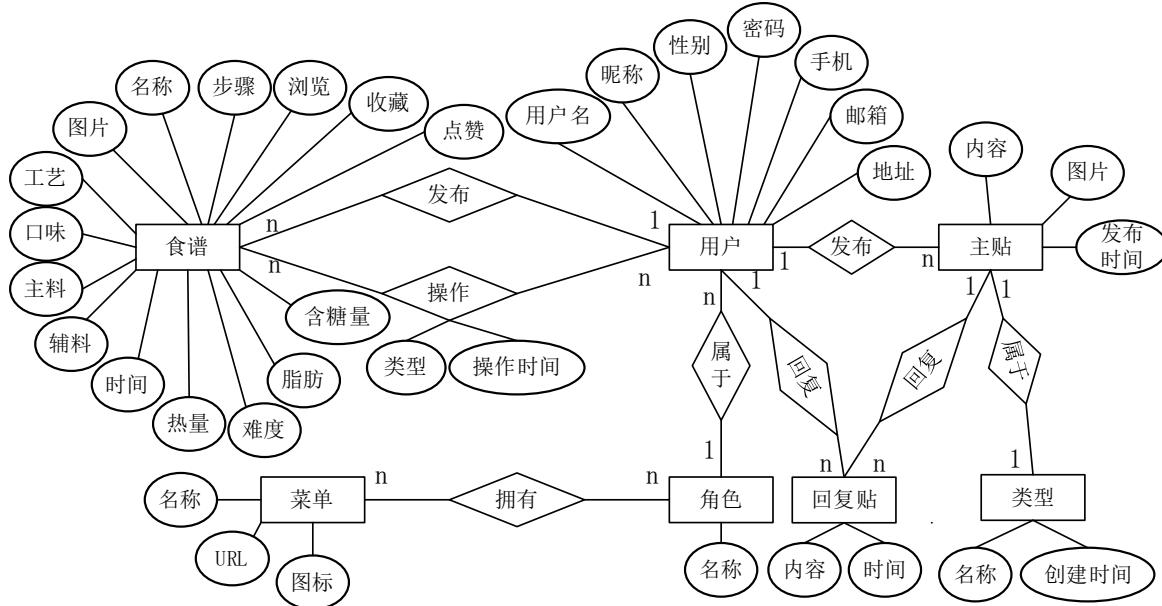


图 4-5 系统 E-R 图

如图 4-5 所示为本系统实体关系图。本系统共有 7 个实体，分别为用户实体、食谱实体、主贴实体、主贴类型实体、回复贴实体、角色实体和菜单实体。

#### 4.4.2 数据库表设计

上一节根据需求分析和功能模块划分设计出完整的 E-R 图模型，在本节将根据 E-R 图模型中实体和实体间的关系设计出具体的数据库表结构。MySQL 数据库中一共有 9 张数据表，分别为用户表 t\_user，食谱表 t\_recipe，食谱操作表 t\_recipe\_op，主贴表 t\_topic，主贴回复表 t\_topic\_reply，主贴类型 t\_topic\_type，菜单表 t\_menu，角色表 t\_role，菜单角色关联表 r\_role\_menu。

(1) **用户表 t\_user** 用户表存储了系统用户和管理员的个人信息，包含用户 id、用户名、密码、性别、年龄、电话、邮箱、地址、头像 URL、座右铭、喜好类别、角色 id 字段，其中用户 id 是唯一的主键，用于唯一标识用户，以下每个表中都会有 id 字段用以唯一标识一条记录。用户表结构如表 4-1 所示。

表 4-1 用户表 (t\_user)

字段名称	数据类型	是否非空	描述
id	int	非空	用户的 id (主键)
username	varchar(22)	非空	用户名
password	varchar(80)	非空	密码 (加密后)

表 4-1 (续表)

字段名称	数据类型	是否非空	描述
name	varchar(20)	可空	昵称
sex	varchar(1)	可空	性别 0:男 1:女
age	tinyint	可空	年龄
phone	varchar(11)	可空	电话
email	varchar(30)	可空	邮箱
address	varchar(100)	可空	地址
face	varchar(30)	可空	头像的 URL
signature	text	可空	座右铭
favor	text	可空	喜好类别（空格分隔）
create_time	datetime	可空	注册时间
login_time	datetime	可空	最后登录时间
role_id	int	可空	角色 id（外键）

(2) 食谱表 t\_recipe 食谱表用来存放爬取的食谱数据，包含食谱 id、食谱名称、图片、工艺、口味、主料、辅料、制作步骤、热量等字段。其中，图片字段主要存放多张图片的 URL，每张图片的 URL 以“@”符号进行分隔，方便后续的数据分离操作。特别的，制作步骤字段中含有多个步骤，每个步骤既存在文字内容，也存在图片，所以每个步骤以“@”进行分隔，某一步骤中文字与图片以“||”进行分隔。这样，在使用这些数据时，只需要对字符串进行以分隔符进行分隔，则会得到单独的文字内容和图片。如表 4-2 所示为食谱表结构设计。

表 4-2 食谱表 (t\_recipe)

字段名称	数据类型	是否非空	描述
id	int	非空	食谱 id（主键）
name	varchar(20)	非空	食谱名称
img	varchar(60)	非空	图片 URL（“@”分隔）
craft	varchar(20)	非空	工艺
taste	varchar(1)	非空	口味
main_ingredient	tinyint	非空	主料（以空格分隔）
sub_ingredient	varchar(11)	非空	辅料（以空格分隔）
step	varchar(30)	非空	制作步骤

表 4-2 (续表)

字段名称	数据类型	是否非空	描述
energy	varchar(100)	可空	热量
sugar	varchar(30)	可空	含糖量
fat	text	可空	脂肪含量
need_time	text	非空	所需时间
level	int	非空	难度
fit_person	varchar(10)	可空	适宜人群
star_num	int	可空	点赞数量
collect_num	int	可空	收藏数量
browse_num	int	可空	浏览数量
create_time	datetime	可空	创建时间
publish_user_id	int	非空	发布人 id (外键)
create_time	datetime	非空	创建时间 (创建时生成)
update_time	datetime	非空	更新时间 (修改时更新)

(3) 食谱操作表 t\_recipe\_op 食谱操作表用来存储用户对食谱的操作数据，包含编号 id、用户 id、食谱 id、操作类型、创建时间。其中，操作类型字段 0 代表点赞，1 代表收藏。如表 4-3 所示为食谱操作表结构设计。

表 4-3 食谱操作表 (t\_recipe\_op)

字段名称	数据类型	是否非空	描述
id	int	非空	食谱操作 id (主键)
user_id	int	非空	用户 id (外键)
recipe_id	int	非空	食谱 id (外键)
type	varchar(1)	非空	类型 (0: 点赞 1: 收藏)
create_time	datetime	非空	创建时间 (创建时生成)
update_time	datetime	非空	更新时间 (修改时更新)

(4) 角色表 t\_role 角色表用来存储系统不同的角色。

表 4-4 角色表 (t\_role)

字段名称	数据类型	是否非空	描述
id	int	非空	角色 id (主键)
name	varchar(20)	非空	角色名称

表 4-4 (续表)

字段名称	数据类型	是否非空	描述
create_time	datetime	非空	创建时间
status	varchar(1)	非空	角色状态
create_time	datetime	非空	创建时间（创建时生成）
update_time	datetime	非空	更新时间（修改时更新）

(5) 菜单表 t\_menu 菜单表主要用于存放管理员后台系统每一个菜单的详细信息及其菜单等级，对于实体关系图中菜单实体，其中包含有菜单 id、名称、URL 地址、排序等级、icon 显示图标、父节点 id、权限状态等字段信息。其中，权限 id 为菜单表主键，用来标识唯一菜单信息，排序等级表示菜单的优先级，如排序等级为 1 则说明该菜单为父节点菜单。菜单表结构设计如表 4-5 所示。

表 4-5 菜单表 (t\_menu)

字段名称	数据类型	是否非空	描述
id	int	非空	菜单 id（主键）
name	varchar(20)	非空	菜单名称
URL	varchar(20)	非空	菜单地址
level	tinyint	非空	排序等级（一级菜单为 1）
icon	varchar(20)	非空	icon 名称
status	varchar(1)	非空	0: 禁用 1: 启用
f_id	int	可空	父节点（自我外键）
create_time	datetime	非空	创建时间（创建时生成）
update_time	datetime	非空	更新时间（修改时更新）

(6) 主贴类型表 t\_topic\_type 主贴类型表用于存储发布的主贴类型，对应了实体关系图中主贴类型实体。其中包含类型 id、名称、状态码、创建时间和修改时间字段；类型 id 为主键来唯一标识主贴类型；状态码字段 0 代表禁用，1 代表启用。主贴类型详细表结构设计如表 4-6 所示。

表 4-6 主贴类型表 (t\_topic\_type)

字段名称	数据类型	是否非空	描述
id	int	非空	类型 id（主键）
name	varchar(20)	非空	名称
status	varchar(1)	非空	0: 禁用 1: 启用

表 4-6 (续表)

字段名称	数据类型	是否非空	描述
create_time	datetime	非空	创建时间(创建时生成)
update_time	datetime	非空	更新时间(修改时更新)

(7) 主贴表 t\_topic 主贴表用于存储用户所发布的主贴信息，对应了实体关系图中主贴实体。其中包含主贴 id、标题内容、正文内容、图片 URL、发布人 id、类型 id、点赞量、浏览量、状态、创建时间和修改时间。主贴 id 用来唯一标识主贴信息，图片 URL 字段中以“@”符号进行分隔来存放多张图片路径，状态字段用来标识主贴的热度类型，如热门、精华、置顶等。如表 4-7 所示为主贴表结构设计。

表 4-7 主贴表 (t\_topic)

字段名称	数据类型	是否非空	描述
id	int	非空	主贴 id(主键)
title	text	非空	标题
content	text	非空	内容
img	text	可空	图片 URL, 以“@”分隔
publish_user_id	int	非空	发布人 id
type_id	int	非空	类型 id
status	varchar(20)	可空	标识主贴是否置顶等状态
create_time	datetime	非空	创建时间(创建时生成)
update_time	datetime	非空	更新时间(修改时更新)

(8) 主贴回复表 t\_topic\_reply 主贴回复表用于存储用户所回复主贴的内容，也用来存储回复其他用户的回复内容。其中包含回复内容 id、具体内容、图片、发布人 id、主贴 id、创建时间、修改时间字段。内容 id 用于唯一标识主贴回复表中的一条内容，主贴 id 为主贴回复表的外键。用户修改当前的回复时，数据库会自动更新修改时间字段。如表 4-8 为主贴回复表结构设计。主贴回复表还存储了本条回复内容所回复的内容的 id。

表 4-8 主贴回复表 (t\_topic\_reply)

字段名称	数据类型	是否非空	描述
id	int	非空	回复 id(主键)
content	text	非空	内容
publish_user_id	text	非空	发布人 id(外键)

表 4-8 (续表)

字段名称	数据类型	是否非空	描述
topic_id	text	非空	回复的主贴 id
last_reply_id	int	非空	上一条回复内容的 id
create_time	datetime	非空	创建时间（创建时生成）
update_time	datetime	非空	更新时间（修改时更新）

(9) 角色菜单关联表 r\_role\_menu      用户角色关联表用于存储各个角色拥有哪些菜单权限，包含主键 id、角色 id、菜单 id 和创建时间。在进行存储时，该表只存储最后一级的菜单信息，这样可以很方便在前端进行控制权限菜单的生成，避免因为加入父级菜单而导致最终生成的菜单不准确的问题。

表 4-9 角色菜单关联表 (r\_role\_menu)

字段名称	数据类型	是否非空	描述
id	int	非空	id (主键)
role_id	int	非空	角色 id (外键)
menu_id	int	非空	菜单 id (外键)
create_time	datetime	非空	创建时间

## 4.5 本章小结

本章主要对本系统进行了详细的总体设计。首先，根据上一章的需求分析结果，先后进行了系统逻辑架构设计和系统技术架构设计，并分别进行了用户端系统功能模块划分和管理员系统功能模块划分，给出了具体的功能模块图。在系统逻辑架构中，本系统自上而下被分为用户层、表现层、控制层、业务层和数据层，各个层级之间各司其职，分担不同的角色和功能。在技术架构中，本系统涉及到前端技术栈和后端技术栈，本章对其中的关键技术作了简要说明。最后，本章完成了系统的功能模块划分，绘制了 E-R 图，并将其转化为具体的表结构。本章内容将为后续的详细设计和编码测试奠定基础。

## 第5章 基于知识图谱的饮食推荐

### 5.1 RippleNet

RippleNet 又名为水波模型，其关键思想是兴趣传播。对于每一个用户，RippleNet 模型将用户的直接交互记录作为知识图谱的种子集合（seed set），沿着知识图谱上路径不断向外扩展和发散，从而形成用户在知识图谱中用户兴趣扩散的集合。

#### 5.1.1 RippleNet 介绍

RippleNet 使用了联合学习的方法，将知识图谱的原始数据作为输入，充分利用知识图谱向外扩散的特性，将用户历史兴趣数据在知识图谱中像水波纹一样不断向外扩散，最终得到扩散后的历史兴趣集合（RippleSet）。RippleNet 将知识图谱的损失函数和模型训练时产生的损失函数进行计算，更新模型的参数，最终得到基于知识图谱的推荐模型。该模型构造了物品  $V$  和用户  $U$  的特征向量表示，计算每个用户感兴趣的 item 的概率，并通过计算得到的用户特征向量预测用户  $U$  对物品  $V$  的感兴趣指标。模型的主要任务和目标是通过用户  $U$  的历史兴趣预测该用户对物品的偏好程度，训练的预测模型为：

$$y_{uv} = F(u, v, \delta) \quad (5-1)$$

在公式(5-1)中， $y_{uv}$  表示用户  $u$  对物品  $v$  的兴趣程度概率， $\delta$  表示输入到模型的其它参数。RippleNet 采用小批量梯度下降的方式进行整体的模型训练，模型的损失函数主要有三部分组成：知识图谱的损失，模型训练时产生的损失，即用户交互矩阵真实值与 RippleNet 模型计算得到的预测值之间产生的交叉熵损失，以及防止模型出现过拟合的正则项。

$$\begin{aligned} \min \mathcal{L} &= -\log((p(Y|\Theta, \mathcal{G}) \cdot p(\mathcal{G}|\Theta))) \\ &= \sum_{(u,v) \in Y} -\left(y_{uv} \log \sigma(u^T v) + (1-y_{uv}) \log(-\sigma(u^T v))\right) \\ &\quad + \frac{\lambda_2}{2} \sum_{r \in R} \|I_r - E^T R E\|_F^2 + \frac{\lambda_2}{2} \left(\|V\|_F^2 + \|E\|_F^2 + \sum_{r \in R} \|R\|_F^2\right) \end{aligned} \quad (5-2)$$

随机采样是推荐系统和机器学习中一种常见的采样方法。RippleNet 模型一共使用了四次随机采样的思想。第一次在对用户与物品交互数据处理时，采用了随机采样

对缺少负样本的用户数据进行采样，随机选择物品作为用户的负样本数据。第二次是在划分训练数据集和测试数据集时进行随机采样划分，确保训练通用性。第三次则是构建用户在知识图谱中的 RippleSet 数据集时，利用随机采样的方式随机获取固定邻居大小的样本。第四次是在 RippleNet 模型训练的过程中使用随机采样的小批量样本代表整体样本进行模型训练，改善了模型训练的运行效率，节省了计算资源。

### 5.1.2 用户兴趣建模

RippleNet 模型的核心思想是利用用户直接交互记录进行数据建模。该模型将用户的直接交互记录作为模型扩散的种子集合，即一个波纹的中心，并将用户的交互记录沿着知识图谱的路径向外展开，即通过波纹中心向外扩散而形成的涟漪集合，将这个集合作为用户扩散后的兴趣集合，即 RippleSet。RippleNet 是一种基于知识图谱的推荐系统模型，利用了多跳键-值注意力机制<sup>[2]</sup>。该机制可以看作为具有多层次的感知机结构，感知机的每一层都有一组键-值对和一个查询向量  $q$ ，通过该查询向量  $q$  和键值对之间的相似度得到相关兴趣的权重向量，并将该权重向量作为下一层感知机的查询向量。通过以上步骤的层层迭代，从而获得用户潜在兴趣的集合。用户和用户的历史兴趣都使用了 Embedding 向量表示，每位用户会通过 Ripple 模型初始化一个查询向量  $q$ ，每个物品也会通过模型初始化 Embedding 向量集合，并将该 Embedding 集合作为物品的值向量。在进行每一次扩散时，RippleNet 使用用户初始化的查询向量  $q$  和物品值向量，计算出加权和作为下一次迭代新的查询向量，最终经过多次迭代将这些查询向量进行拼接，最终得到 MLP（多层次感知机 Multi-Layer Perception）作为新的推荐结果。RippleNet 模型的多跳注意力机制能够在保证推荐结果的同时，减少模型的计算量和模型参数量，能够很好展示用户兴趣在知识图谱上的传播过程。

RippleNet-CA 模型是基于 RippleNet 改进的推荐模型，对 RippleNet 模型知识图谱的随机采样方法进行了改进，将用户的历史兴趣在知识图谱上扩散传播时时采用的固定邻居个数随机采样的方法，修改为基于知识图谱扩散偏好集合共有属性的随机采样方法，提高了采样集合包含的用户兴趣的信息量，很好解决了数据稀疏问题。

用户  $U$  的第  $k$  跳的波纹集合被定为头节点  $h$  在  $\varepsilon_u^{k-1}$  中的尾节点的集合：

$$\varepsilon_u^k = \{t | (h, r, t) \in \mathcal{G} \text{ and } h \in \varepsilon_u^{k-1}\}, k = 1, 2, \dots, H \quad (5-3)$$

如公式(5-3)所示，当  $k=0$  时， $\varepsilon_u^0$  表示用户当前收藏的食谱序列。当  $k=1$  时，表示以用户收藏的序列为头实体的三元组的尾实体。其他情况依次类推，可以得到  $\varepsilon_u^k$  就

是第  $k$  个波纹上的实体。如上述可得，用户  $u$  的第  $k$  跳的波纹集合的三元组为：

$$S_u^k = \{(h, r, k) | (h, r, t) \in \mathcal{E}_u^{k-1}\}, k = 1, 2, \dots, H \quad (5-4)$$

RippleNet 的用户特征向量结合了多个节点特征，并且根据计算结果所得的相关性给出推荐结果，具有较强可解释性。 $h$  和  $v_i$  在关系空间  $R_i$  的相似度叫做相关概率  $p_i$ ，需要使用的是 1-hop（知识图谱中第一次向外扩展的食谱信息）的 RippleSet，对于 set 中的每一个  $(h, r, t)$ ，用  $(h * r)$  与 item  $v$  相乘得到 item  $v$  与 1-hop 里每个  $(h_i, r_i)$  的相关性得分，再通过 softmax 进行相关性得分的归一化，计算公式如公式(5-5)所示：

$$p_i = \text{softmax}(v^T R_i h_i) = \frac{\exp(v^T R_i h_i)}{\sum_{(h, r, t \in S_u^1)} \exp(v^T R_i h_i)} \quad (5-5)$$

用户的兴趣特征  $u$  的计算公式如公式(5-6)所示：

$$u = \sum p_i t_i \quad (5-6)$$

最终，结合计算的用户特征和物品特征得出预测的物品点击概率：

$$y_{uv} = \text{sigmoid}(u^T v) \quad (5-7)$$

由于本项目中的知识图谱中食谱指向主料、辅料等属性，而主料并非指向食谱，所以需要规定偏好传播过程中的路径实体类型。用户兴趣在构建的食谱知识图谱传播过程中，奇数跳尾的节点代表了用户感兴趣食谱的属性实体（如工艺、口味等），偶数跳尾的节点代表了偏好属性对应的食谱实体。

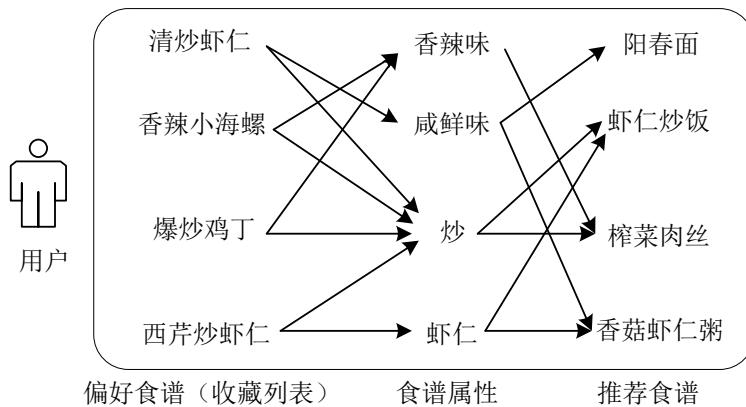


图 5-1 用户感兴趣食谱共有属性

如图 5-1 所示，一跳的波纹集合用户偏好的属性实体为：香辣味、咸鲜味、炒和虾仁等。二跳波纹集合的用户扩散偏好食谱实体为：阳春面、虾仁炒饭、榨菜肉丝和香菇虾仁粥。每个通过传播后的食谱属性节点在提取出的三元组中的  $(r, t)$  组合特征共有频次代表了用户对该偏好食谱的喜爱程度。

用户兴趣经过两跳以上的传播，由于相关概率  $p_i$  小于 1，三跳及以后的偏好与种

子相关性急剧下降且 RippleSet 的数量呈爆炸级别增长，增加了额外工作量。所以，此项目主要选择跳数  $H=2$ 。

### 5.1.3 共有属性采样算法

RippleNet 模型在用户偏好扩散时采用了固定邻居个数随机采样的方法，可能在波纹扩散的过程中出现重要信息丢失的现象。RippleNet-CA 模型提出了 RippleSet 共有属性采样来重新构建用户的兴趣特征  $u$ ，以构建推荐算法。如表 5-1 所示，为共有属性采样数据增强算法。通过在知识图谱中沿着路径多跳对食谱数据共有属性采样，可以获得用户的偏好扩散，并将之作为用户的历史兴趣对用户偏好数据集的增强。

表 5-1 共有属性采样数据增强算法

---

#### 算法 5.1 共有属性采样数据增强算法

---

**输入：**交互记录  $\mathbf{Y}$ 、知识图谱  $\mathbf{G}$ 、参数  $k$

**输出：**交互记录  $\mathbf{Y}$

1. 在知识图谱上为每位用户随机采样波纹集合  $\text{RippleSet}\{s_u^k\}_{k=2}^H$
  2. **for**  $(h, r, t)$  in  $\{s_u^k\}_{k=2}^H$  **do**
  3.     统计不同  $h$  共有的  $(r; t)$  类别数  $m$  和频数  $n_{uc}$
  4.     **if**  $n > k$  **and**  $t$  不在交互记录  $\mathbf{Y}$  的正样本中 **then**
  5.         将  $t$  加入交互记录  $\mathbf{Y}$  正样本中
  6.     **else if**  $n = 0$  **and**  $t$  不在交互记录  $\mathbf{Y}$  的负样本中 **then**
  7.         将  $t$  加入交互记录  $\mathbf{Y}$  负样本中
  8.     **end for**
  9. **return**  $\mathbf{Y}$
- 

## 5.2 数据集的获取与处理

### 5.2.1 数据集的获取

本文是基于知识图谱的饮食推荐系统，本系统所使用的数据集包含饮食知识图谱数据和用户-食谱偏好数据，这些数据均从美食杰网站进行爬取，使用的工具由 PyCharm、Python、Scrapy、Xpath、Excel 等。爬虫数据使用了 Scrapy 框架，采用了可读性较强的 Xpath 代替正则表达式。Scrapy 框架可以异步在不同的 URL 上爬取数

据，并通过管道的方式存储在指定的文件中。

通过调研美食杰网站数据分布时，制定了两种爬取数据方案。方案一是根据美食杰网站所列举的大类和小类为主要切入点，爬取每个类别下的食谱数据。方案二是直接爬取用户的收藏列表食谱数据，得到用户-食谱偏好数据集，再通过去重的方式获得食谱数据。

	A 食谱名称	B 计数项：食谱名称
1	可乐鸡翅	17
2	糖醋排骨	15
4	宫保鸡丁	14
5	鱼香肉丝	13
6	酸菜鱼	12
7	水煮肉片	12
8	鱼香茄子	11
9	水煮牛肉	10
10	糖醋里脊	10
11	麻婆豆腐	10
12	红烧排骨	10
13	红烧肉	10
14	油焖大虾	9
15	清蒸鲈鱼	8
16	梅菜扣肉	8
17	蚂蚁上树	8
18	红烧茄子	8
19	红烧带鱼	8
20	洋葱炒鸡蛋	7
21	凉拌黄瓜	7
22	回锅肉	7
23	干煸四季豆	7
24	西红柿炒鸡蛋	6

图 5-2 重复食谱数据统计图

如图 5-2 所示，为方案一爬取家常菜类别食谱的部分结果，其中出现了多个高重复食谱数据，也存在不属于家常菜类别的食谱，由此可得出美食杰网站的大类和小类的划分不准确，食谱分布不清晰，所以不采取方案一。

方案二需要遍历所有的用户列表，再通过用户列表进入用户主页爬取该用户所收藏的食谱信息，最后通过这些食谱信息去掉重复值可得食谱信息。美食杰的用户列表页面在访问的 URL 的 page 参数中，程序可以通过提前设置需要爬取用户列表的页数范围，构造相应的用户列表 URL 地址数组，然后再依次进入用户主页获取收藏列表，并进行获取食谱数据。具体的数据爬取步骤如下所示：

- (1) 通过提前设置的页码范围构造用户列表的 URL 地址数组；
  - (2) 从这些用户列表网页中解析用户的 id，构造用户的 URL 地址；
  - (3) 依次访问 (2) 中所有用户主页的 URL；
  - (4) 从 (3) 中进入的用户主页中获取用户的个人信息和该用户收藏列表的食谱 id 及详情页的 URL 地址，并将食谱详情页地址加入待爬取 URL 地址列表中；
  - (5) 对 (4) 中获取的食谱网页进行解析，得到食谱数据，并存入 Excel 表格中；
- 美食杰网站的数据只需一次加载，便全都渲染在页面的标签中，无需通过点击按

钮等操作动态获取数据，所以在食谱详情页直接通过 XPath 方式获取 HTML 中渲染的数据即可。

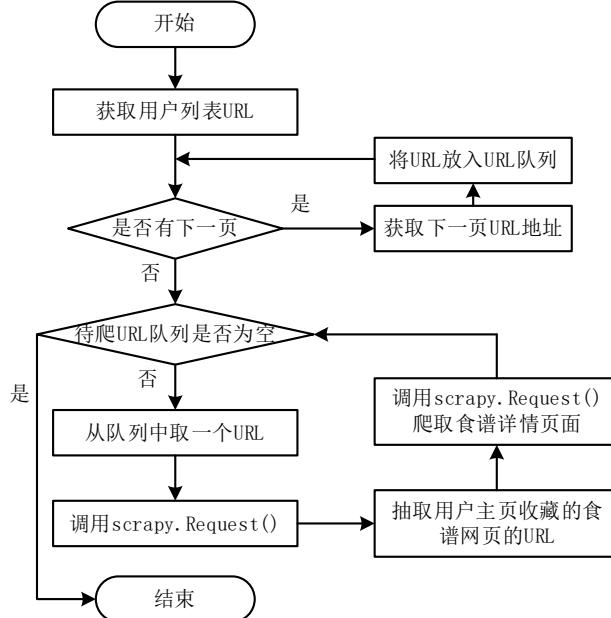


图 5-3 Scrapy 爬虫流程图

如图 5-3 所示，为爬取数据流程图。首先，先获取所有用户列表的网址集合，再依次遍历获取用户的主页网址，将其加入到待爬取 URL 队列中。访问每位用户的主页地址，依次循环遍历用户的收藏食谱列表，获取该用户收藏食谱 id 和详情页 URL 地址，并继续进入食谱详情页面获取食谱的详细数据。如图 5-4 所示，为爬取的美食杰网站部分原始结果。

A	B	C	D	E	F	G	H	I	J	K	L	M
name	img	craft	taste	main_ingredient	sub_ingredient	step	energy	sugar	fat	need_time	level	fit_person
1 木樨豌豆	https://st-cn.me炒	家常味	鸡蛋140克 豌豆(鲜)100克 花生油20毫升 盐少量 麻	准备食材：鸡蛋140克 豌豆(鲜)100克 花生油20毫升 盐少量 麻		准备食材：鸡蛋140克 豌豆(鲜)100克 花生油20毫升 盐少量 麻	258.5	18.41	12	5分钟	初级入门	1.21.11
2 南瓜小米粥	https://st-cn.me煮	甜味	老南瓜200克 小米100克 水适量	准备老南瓜和小米200克 小米100克 水适量		准备老南瓜和小米200克 小米100克 水适量	272.5	10.92	0.5	60分钟	新手尝试	2.00.00
3 柠香葱油焗鸡腿	https://st-cn.me焗	葱香味	鸡腿2个 葱油2勺 酱油1勺 海盐适量 姜片5片 蒜5片	准备好鸡腿，清油6.3		准备好鸡腿，清油6.3	256.3	1.75	4.3	60分钟	新手尝试	0.00.00
4 土豆烧鸡腿块	https://st-cn.me炒	家常味	鸡腿3只 土豆300克 洋葱2颗 食用油适量 生抽适量 料	切好各材料、鸡腿3只 土豆300克 洋葱2颗 食用油适量 生抽适量 料		切好各材料、鸡腿3只 土豆300克 洋葱2颗 食用油适量 生抽适量 料	286.6	7.49	2.8	15分钟	未知	0.00.11
5 烤烧鸡腿	https://st-cn.me烤	咸鲜味	鸡腿3根	酱油1勺 白砂糖1勺 盐15克 鸡腿洗净，去鸡皮113.9		酱油1勺 白砂糖1勺 盐15克 鸡腿洗净，去鸡皮113.9	257.8	3.78	4.5	15分钟	新手尝试	0.01.00
6 青尖椒土豆烧鸡	https://st-cn.me烧	香辣味	青尖椒500克 土豆3-5个	油适量 香料少许 回香少许 桂皮、鸡翅洗净后切成166		油适量 香料少许 回香少许 桂皮、鸡翅洗净后切成166	32.2	10.15	7	60分钟	初级入门	0.01.21
7 土豆炖茄子酱	https://st-cn.me炖	酱香味	茄子2个 土豆2个	油适量 盐适量 五香粉适量 茄子在淡盐水浸32		油适量 盐适量 五香粉适量 茄子在淡盐水浸32	32.2	4.9	0.2	10分钟	初级入门	0.00.00
8 牛肉炖土豆	https://st-cn.me炖	咸鲜味	牛腩500克 土豆2个 胡萝卜	洋葱适量 大葱1小段 香葱牛腩清洗干净切19.3		洋葱适量 大葱1小段 香葱牛腩清洗干净切19.3	182.9	8.61	0.2	90分钟	初级入门	1.00.01
9 新奥尔良烤翅	https://st-cn.me烤	香辣味	鸡翅中500克	盐2克 蜂蜜2汤匙 清水少		盐2克 蜂蜜2汤匙 清水少	212.4	3.85	6.4	15分钟	初级入门	1.00.00
10 苹果红枣红糖水	https://st-cn.me煲	甜味	苹果一个 红枣适量 枸杞15克 红糖半斤	准备食材@https:65.7		准备食材@https:65.7	212.4	11.27	0.2	2小时	初级入门	0.20.00
11 凉拌香辣海带丝	https://st-cn.me拌	香辣味	海带丝一份	小米辣六七个【根据自己今天给你们看看才57.2		小米辣六七个【根据自己今天给你们看看才57.2	212.4	6.44	2.1	30分钟	初中水平	1.01.01
12 红烧猪蹄	https://st-cn.me炒	家常味	猪蹄	葱 姜 白芷2片 桂皮1块 猪蹄焯水，放入煮156		葱 姜 白芷2片 桂皮1块 猪蹄焯水，放入煮156	212.4	6	11.3	5分钟	新手尝试	1.11.00
13 红枣糕	https://st-cn.me烤	甜味	红枣100克	红枣80克 玉米油100克 小红枣已经下锅煮，349.1		红枣80克 玉米油100克 小红枣已经下锅煮，349.1	212.4	24.36	20.8	30分钟	未知	1.22.11
14 电饭锅版排骨焖饭	https://st-cn.me焖	家常味	排骨300克 红萝卜半根 干耗油2勺 生抽2勺 姜丝少	配料图@https://s/173		配料图@https://s/173	212.4	5.18	12.4	60分钟	初中水平	1.01.10
15 杭椒牛柳	https://st-cn.me炒	黑椒味	牛里脊肉200克 杭椒6根	生抽15毫升 老抽10毫升 准备牛里脊200克101.7		生抽15毫升 老抽10毫升 准备牛里脊200克101.7	212.4	5.25	2.6	10分钟	初中水平	0.10.10
16 杭椒牛柳	https://st-cn.me炒	家常味	嫩牛肉200克 杭椒200克	蚝油适量 糖适量 生抽适量 准备好所用食材		蚝油适量 糖适量 生抽适量 准备好所用食材	7.02	2.38	2.2	5分钟	新手尝试	0.00.01
17 山楂酱	https://st-cn.me煮	酸甜味	山楂200克	柠檬汁5毫升 水适量 冰糖山楂洗净，可先打7.5		柠檬汁5毫升 水适量 冰糖山楂洗净，可先打7.5	212.4	3.37	0.5	30分钟	新手尝试	2.00.00
18 什锦蛋炒饭	https://st-cn.me炒	咸鲜味	米饭300克 鸡蛋2个	培根适量 洋葱适量 胡萝卜适量 提前把饭煮好，120.7		培根适量 洋葱适量 胡萝卜适量 提前把饭煮好，120.7	212.4	11.48	3.5	15分钟	新手尝试	1.20.00
19 红豆酥	https://st-cn.me烘焙	甜味	红豆10个	蛋挞皮10个 红豆200克 白糖适量		蛋挞皮10个 红豆200克 白糖适量	182.9	14.49	5.6	30分钟	新手尝试	0.12.00
20 菠菜蒸蛋	https://st-cn.me蒸	家常味	鸡蛋3个 菠菜3棵	盐1克 生抽2勺 蒜泥适量 菠菜洗净去根，7.84		盐1克 生抽2勺 蒜泥适量 菠菜洗净去根，7.84	182.9	0.84	5.6	10分钟	新手尝试	0.00.00
21 油爆皮皮虾	https://st-cn.me炒	家常味	皮皮虾650克 生姜2片 蒜5瓣	盐适量 鸡精少许		盐适量 鸡精少许	182.9	0.56	0.77	5分钟	新手尝试	0.00.00
22 羊肉粉丝汤	https://st-cn.me煮	家常味	冻羊肉片200克 娃娃菜5片 盐适量	皮皮虾清洗干净，7.3		皮皮虾清洗干净，7.3	182.9	6.44	1.4	15分钟	新手尝试	0.10.01
23 板栗烧排骨	https://st-cn.me烧	家常味	排骨400克 板栗200克	油20克 糖10克 生抽1勺		油20克 糖10克 生抽1勺	161.2	7.42	10.4	60分钟	初级入门	1.10.11
24 南瓜饼	https://st-cn.me炒	家常味	南瓜300克 糯米粉300克	白糖1勺 面包糠		白糖1勺 面包糠	182.9	17.43	0.3	5分钟	新手尝试	2.10.00
25 芝士焗饭	https://st-cn.me烤	酸甜味	米饭1碗 肉末50克	番茄酱20克 鸡蛋1个 马铃薯饭一碗。番茄酱152.4		番茄酱20克 鸡蛋1个 马铃薯饭一碗。番茄酱152.4	182.9	13.58	6	10分钟	新手尝试	0.12.00
26 豆腐炒白菜	https://s.1ig.mei炒	咸鲜味	毛豆腐250g 白菜120g	食盐适量 酱油适量 葱适量		食盐适量 酱油适量 葱适量	159.9	2.45	2.6	未知	未知	0.00.00
27 鲜汁油菜香菇	https://s.1ig.mei其它1其它口味	其它口味	油菜100g 香菇100g	材料油菜、香菇5.9		材料油菜、香菇5.9	182.9	0.84	0.1	未知	未知	0.00.00

图 5-4 部分网站原始数据结果图

在使用 Scrapy 爬虫框架进行数据获取时，程序会因为数据为 NULL 等问题出现异常和中断情况。为了获得高质量的数据和保证程序稳定运行，程序设置了多处判空

和异常检查的操作，以提高程序的健壮性。在爬取到一条食谱交互数据后，Scrapy 框架通过预设的格式输出添加到 Excel 表格中，最终生成原始网站数据。由于网站自身的限制和安全的考虑，数据爬取设置了较慢的爬取速度，所以在项目实现的过程中，适时开启爬虫程序，以获得更多的原始数据。

### 5.2.2 数据集预处理

如图 5-2 所示，为成功爬取的用户-食谱偏好数据，通过去除重复的食谱数据便可得到食谱数据。美食杰网站的食谱数据集可能存在数据的缺失（工艺、口味类别缺失）、用户上传的食谱数据质量较差（上传步骤少，图片少）、食谱的各个字段描述不完整不规范、食谱名称重复现象严重等问题，本节将对爬取数据进行预处理。

针对食谱数据缺失的问题，本文会首先从数据集中提取数据缺失的食谱信息并进行删除，其次将提取的缺失数据食谱信息通过豆果网、美食天下等网站逐一补齐，并删除无法补齐的无效数据。删除重名食谱的元素，只保留一种。食谱数据有很多字段保留了多个值，均以“@”、“||”等特殊符号进行分隔，方便后续数据的提取和使用。对于“鸡蛋 140 克”、“醋适量”等字段内容，使用正则表达式进行剔除无关信息，保留“鸡蛋”等关键词。数据预处理还有其它处理过程，这些过程均是通过 Excel 软件操作完成，以提高数据集的质量。如表 5-2 所示，为原始数据的数据量统计表。

表 5-2 原始数据数量统计表

食谱实体数量	属性实体数量	关系数量	用户数量	用户-食谱交互数量
5708	5104	80734	1926	55367

### 5.2.3 构建知识图谱

本项目使用了 Neo4j 图数据库存储知识图谱，其中所涉及的工具有 Pycharm、Neo4j Desktop 和 py2neo。食谱主要的信息包括以下几个字段，食谱 id，食谱名，主食材，辅食材，烹饪方式，烹饪时间，烹饪步骤，图片等内容。本项目先将处理好后的食谱信息存入到 Excel 表格中，再通过 Python 程序以 Excel 表格行为单位依次读取每一条食谱信息，将食谱信息各个字段拆分为关系存储到列表中，其中关系有食谱-主食材，食谱-辅食材等。其次，将所有的头节点和尾节点以 Set 的集合形式进行存储，避免出现重复的实体。最后，将这些列表数据依次插入到图数据库中。

在本项目中，饮食知识图谱设计了两种类型的节点：食谱（recipe）节点和食谱属性节点（recipe\_attribute）。食谱节点个数与爬取的食谱数一致，食谱属性节点分为以下 6 类：主料（main\_ingredients），辅料（sup\_ingredients），口味（taste），工艺（craft），所需时间（need\_time），难度等级（level）。设计了 6 种关系类型：食谱的主料（recipe\_maingredients），食谱的辅料（recipe\_supingredients），食谱的口味（recipe\_taste），食谱的工艺（recipe\_craft），食谱的所需时间（recipe\_need\_time），食谱的难度等级（recipe\_level）。

5688	5687	东北大饭包	5084	10792	炸 craft	5688	5687	东北大饭包
5689	5688	奶黄包	5085	10793	冻 craft	5689	5688	奶黄包
5690	5689	牛肉拉面	5086	10794	蒸 craft	5690	5689	牛肉拉面
5691	5690	剁子	5087	10795	一天 need_time	5691	5690	剁子
5692	5691	减肥菜—凉拌花木耳	5088	10796	数天 need_time	5692	5691	减肥菜—凉拌花木耳
5693	5692	洋葱炒甜椒	5089	10797	未知 need_time	5693	5692	洋葱炒甜椒
5694	5693	蘑菇豆腐瘦肉汤	5090	10798	60分钟 need_time	5694	5693	蘑菇豆腐瘦肉汤
5695	5694	莲藕胡萝卜炖排骨	5091	10799	30分钟 need_time	5695	5694	莲藕胡萝卜炖排骨
5696	5695	过年杀猪菜	5092	10800	2小时 need_time	5696	5695	过年杀猪菜
5697	5696	番茄炖牛腱子	5093	10801	15分钟 need_time	5697	5696	番茄炖牛腱子
5698	5697	花甲粉丝	5094	10802	5分钟 need_time	5698	5697	花甲粉丝
5699	5698	简单快手家常寿司的做法	5095	10803	10分钟 need_time	5699	5698	简单快手家常寿司的做法
5700	5699	油泼辣子	5096	10804	90分钟 need_time	5700	5699	油泼辣子
5701	5700	花开富贵虾—蒜蓉粉丝蒸虾	5097	10805	数小时 need_time	5701	5700	花开富贵虾—蒜蓉粉丝蒸虾
5702	5701	肉末炒豆角	5098	10806	未知 level	5702	5701	肉末炒豆角
5703	5702	鸡蛋虾皮紫菜汤	5099	10807	新手尝试 level	5703	5702	鸡蛋虾皮紫菜汤
5704	5703	蒜香荷兰豆	5100	10808	中级掌勺 level	5704	5703	蒜香荷兰豆
5705	5704	酸辣炝炒藕片	5101	10809	初级入门 level	5705	5704	酸辣炝炒藕片
5706	5705	咖喱土豆鸡块	5102	10810	初中水平 level	5706	5705	咖喱土豆鸡块
5707	5706	速冻饺煎饺	5103	10811	厨神级 level	5707	5706	速冻饺煎饺
5708	5707	自制炒酸奶	5104	10812	高级厨师 level	5708	5707	自制炒酸奶
5709	5708	爆炒花蛤				5709	5708	爆炒花蛤

图 5-5 知识图谱构建文件

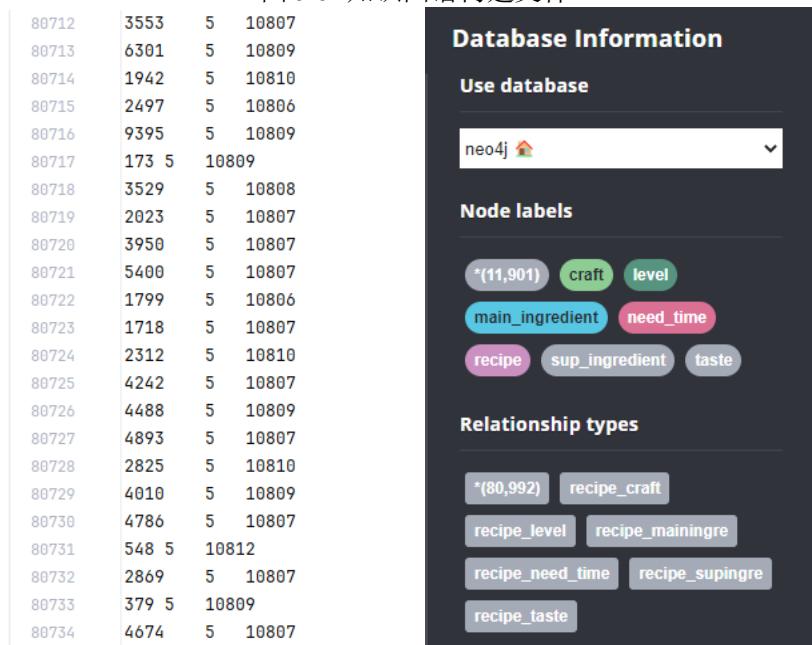


图 5-6 知识图谱构建结果图

如图 5-5 所示，为构建知识图谱的 id 与实体名称对应图，其中包含了食谱实体、主料实体、辅料实体、口味实体、烹饪方式实体、烹饪时间实体、难度等级实体。如

图 5-6 所示, 为知识图谱构建结果图, 对各个实体的关系进行抽离、编号和存储, 并将各个实体和实体间的关系通过 Python 和 py2neo 导入到 Neo4j 图数据库。由图 5-6 可以看出, 导入知识图谱的实体个数有一万多种, 关系个数有八万多种, 其中包含了 7 种实体类型和 6 种关系类型。

### 5.2.4 数据集增强

通过查询用户-食谱偏好数据集得知, 交互记录数为 55367, 用户数为 1926, 食谱数为 5708, 平均每个用户有 20 条交互记录。根据数据稀疏度计算公式计算得到食谱数据集交互密度为 0.3%, 即用户无评分数据占据整个样本空间 99.7%, 要从 0.3% 的用户-食谱交互记录数据学习用户偏好, 数据集太过稀疏, 模型很难学习到用户偏好特征, 模型容易欠拟合。数据集的稀疏主要是由于用户和食谱交互不充分。食谱数据集稀疏度的计算如公式(5-8)所示:

$$\text{数据稀疏度} = 1 - \frac{\text{总交互记录}}{\text{用户数} \times \text{食谱数}} \quad (5-8)$$

用户兴趣在知识图谱的传播过程像水波纹扩散一样, 随着波纹扩散, 波纹逐渐衰减, 但存在多个波源时, 用户的偏好物品扩散出的波纹集会产生多个波纹, 波纹间通过偏好物品间的共有属性产生联系相互干涉叠加形成干涉加强区域, 即下图中深色节点集合。

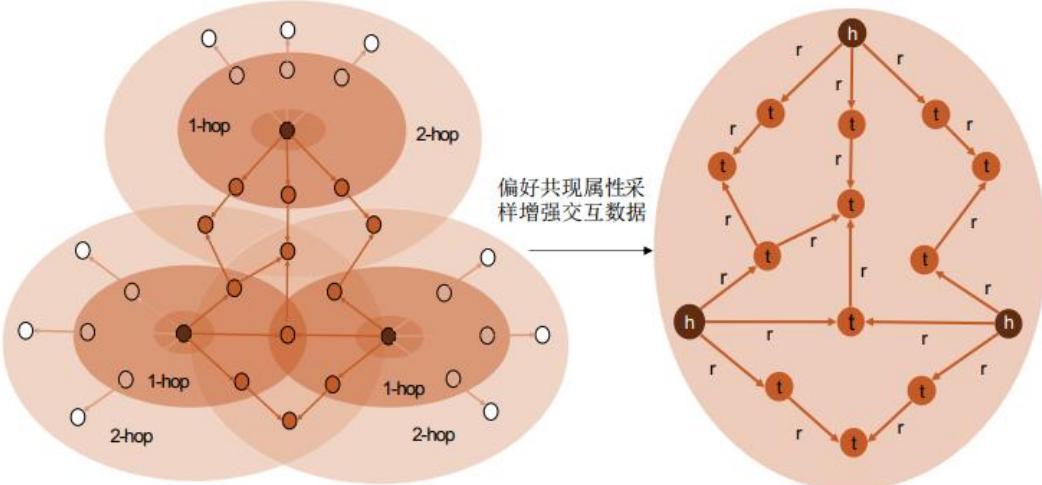


图 5-7 用户兴趣食谱共有属性图

如图 5-7 所示, 用户的偏好数据通过偏好共现属性采样增强后, 可以获得用户的偏好扩展, 并将之看作用户的感兴趣食谱对用户的偏好数据进行了增强。该增强算法如表 5-1 所示。

## 5.3 实验与分析

### 5.3.1 实验环境

本节将分别介绍实验的硬件环境和软件环境。

#### 5.3.1.1 硬件环境

- (1) 显卡为 NVIDIA GeForce GTX 1660Ti 6GB。
- (2) 内存为 16GB。
- (3) CPU 为 Core i7-10750H。

#### 5.3.1.2 软件环境

- (1) 操作系统为 Windows 11。
- (2) Pytorch 版本号为 1.10.2。
- (3) 操作软件为 Pycharm。

### 5.3.2 实验结果及分析

(1) 超参数对比实验      模型的超参数是指固定采样的邻居个数 Size of ripple set, 不同 hop 最大跳数 H, 不同嵌入维度数 D。本实验将使用电影数据集 ml-1m 和本项目构建的食谱数据集 myrecipe 上进行调整实验的超参数进行对比实验。实验结果如表 5-3、5-4、5-5 所示。其中, raw-myrecipe 为原始用户交互数据集, myrecipe 为加强后的用户交互数据集。

表 5-3 不同的固定采样邻居个数对 RippleNet 模型的 AUC 影响

固定采样邻居个数	2	4	8	16	32	64
ml-1m	0.901	0.905	0.912	0.915	<b>0.918</b>	0.916
raw-myrecipe	0.625	0.632	0.634	0.635	<b>0.637</b>	0.636
myrecipe	0.803	0.809	0.816	0.824	<b>0.833</b>	0.830

表 5-4 不同的 hop 跳数 H 对 RippleNet 模型的 AUC 影响

hop 跳数 H	1	2	3	4
ml-1m	0.917	<b>0.920</b>	0.914	0.912
raw-myrecipe	0.636	<b>0.638</b>	0.634	0.633
myrecipe	0.812	<b>0.830</b>	0.826	0.826

表 5-5 不同的嵌入维度数 D 对 RippleNet 模型的 AUC 影响

嵌入维度数 D	2	4	8	16	32	64
ml-1m	0.910	0.917	0.920	<b>0.922</b>	0.917	0.782
raw-myrecipe	0.630	0.634	0.637	<b>0.639</b>	0.632	0.530
myrecipe	0.820	0.824	0.826	<b>0.835</b>	0.830	0.760

通过上述多次的对比实验，可以得出选取 RippleNet 得到最优 AUC 的超参数，其中，固定采样邻居个数为 32，最大的 hop 跳数 H 为 2，向量嵌入维度数 D 为 16。

(2) 数据增强对比实验 数据增强算法在执行的过程中，不断调整参数  $k$  的值，跳数不能太深否则扩展的用户偏好数据与用户的历史兴趣相关度较低，且  $k$  不能太大否则将达不到 ml-1m 的规模， $k$  不能太小否则用户的潜在偏好与用户的相关性较差。经过实验得，当  $k=5$  时，通过数据增强算法扩展得到的用户潜在偏好数据集稀疏度与推荐系统主流的数据集 ml-1m 稀疏度基本相近。

如表 5-6 所示，为数据增强对比实验，将原始的用户历史交互记录通过共有属性采样算法进行数据增强后，RippleNet 模型的 AUC 和 ACC 值有了显著增强。

表 5-6 数据增强对比实验

模型	数据集	AUC	ACC
RippleNet	ml-1m	0.921	0.844
RippleNet	raw-myrecipe	0.636	0.686
RippleNet	myrecipe	<b>0.833</b>	<b>0.827</b>

实验表明，raw-myrecipe 数据集过于稀疏，且分布不均匀，经过数据处理和增强后，myrecipe 数据集的稀疏问题得到一定改善，表现有所提升，但模型仍在本文构建的数据集中表现欠佳，准确率较低。本文中的食谱知识图谱只有六类关系，且尾实体没有考虑食谱实体，导致实体间的交互复杂度比 ml-1m 差，且自建的数据集含有大量未能处理的噪点数据，用户的交互数据不准确等问题。原始数据不存在负样本，采样时使用随机采样的方式，与实际值产生偏差，且数据集过于单一，质量较差，用户与食谱数据交互不充分。

## 5.4 基于 Neo4j 个性化饮食推荐

本项目知识图谱通过 Python 程序、Neo4j 工具半自动构建，使用 Excel 表格的

数据构建知识图谱。图数据库 Neo4j 内置了多种图算法，可以通过节点的入度、出度和介数，挖掘节点的权重，获取节点间的潜在联系，从而得到用户潜在偏好食谱。

**(1) 基于入度排序的食谱推荐** 本项目在使用 Neo4j 构建的知识图谱中，食谱节点的出度表示食谱节点指向属性节点的边的个数。食谱属性节点的入度表示指向同一个属性节点的边的个数。属性节点的入度在一定程度上可以表示该节点在不同食谱上同现的频次，代表了该属性节点在知识图谱中的热度，反映了食谱属性在知识图谱中的偏好分布。

如图 5-8 所示，选取用户 A 收藏的 6 个食谱作为案例进行分析。节点“咸鲜味”的入度为 4，节点“初级入门”的入度为 5，主料节点“虾仁”、“南瓜”、“糯米粉”的入度均为 2，所需时间节点“30 分钟”的入度为 4。通过分析用户收藏的食谱所设计的知识图谱各个节点的入度，可以得出用户上传的食谱共有属性的排名。综上可得，该用户的偏好口味是咸鲜味和甜味，并且喜欢咸鲜味胜过甜味，所以在推荐时推荐这两种口味的食谱给用户并且优先推荐食谱属性中包含入度数大的属性的食谱。

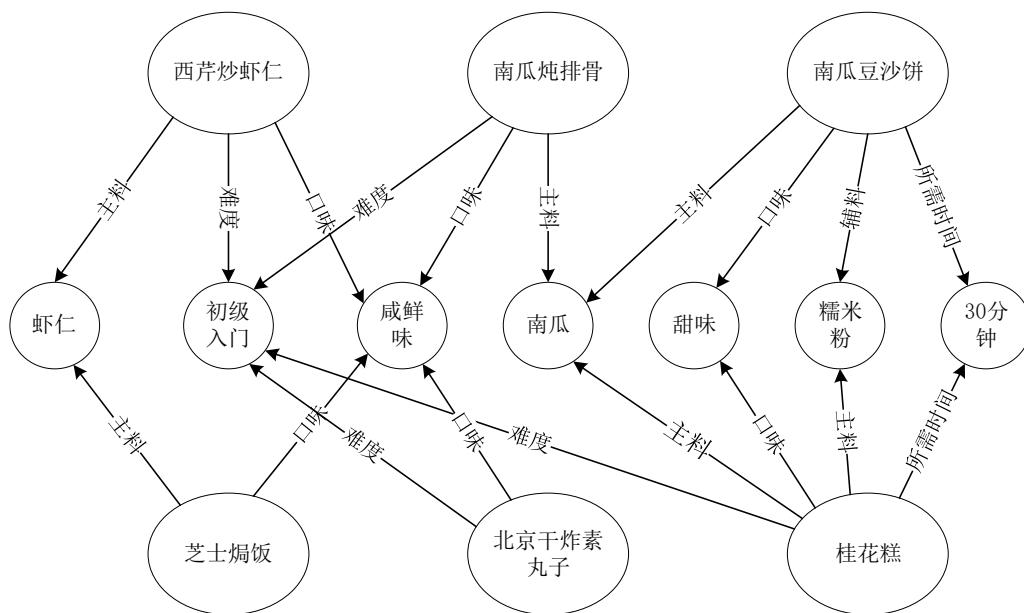


图 5-8 基于入度排名的食谱关系图

表 5-7 所用户食谱口味热度表 (8 项)

node_id	node_labels	node_name	in_degree
11822	taste	家常味	2254
11841	taste	咸鲜味	1524
11823	taste	香辣味	692

表 5-7 (续表)

11835	taste	甜味	502
11821	taste	酸甜味	250
11838	taste	酸辣味	151
11829	taste	酱香味	136
11842	taste	黑椒味	44

从表 5-7 所示可得，美食杰网站的用户上传的食谱，基本反映了中国人的饮食偏好。排在前列的家常味、咸鲜味、香辣味等口味是典型的中式菜系的口味。鸡蛋、面粉、猪肉、五花肉、胡萝卜、西红柿等是食谱中较为常见的食材。通过对属性入度的排序，可以看出用户的饮食偏好分布。

表 5-8 查询知识图谱入度排名前 10 的主食材

Cypher 查询语句

```

MATCH (p:recipe)-[r]->(t:main_ingredient)
WITH t, count(t) as inDegree
ORDER by inDegree DESC
RETURN t.name, inDegree
LIMIT 10

```

如表 5-8 所示，为查询饮食知识图谱入度排名前 10 的主食材的 Cypher 查询语句。通过知识图谱的查询语句，基于节点的入度筛选食谱是一个解决冷启动较好的解决方案。对于新注册到网站的用户，可以通过分析热度较高的食谱或用户感兴趣类别的食谱推荐给用户。通过 Neo4j 的入度出度查询排名，不仅可以获得较为热门的食谱，还可以获得最热门的食材、食谱口味等。

(2) 基于相似度的食谱推荐 使用 Neo4j 数据库内置的 NPOC 插件计算节点间的文本相似度可以挖掘食谱节点间的潜在联系，得到用户推荐食谱结果。

表 5-9 查询与可乐基础食材相似度排名前 10 的食谱

Cypher 查询语句

```

WITH "可乐鸡翅" AS target_name
MATCH(h1:recipe{name:target_name})-[r1]->(t1:main_ingredient)<-[r2]-(h2:recipe)-
[r3]->(t2:main_ingredient)

```

```
WHERE r1<>r2  
WITH p2,  
    apoc.text.levenshteinDistance(t1.name, t2.name) AS dist  
ORDER BY dist ASC  
RETURN h2.name AS recipe, distance LIMIT 10
```

---

如表 5-9 所示为查询语句，通过匹配从食谱“可乐鸡翅”节点出发经过食谱属性节点“t1”，经过第二个食谱节点“p2”，最后指向第二个食谱属性节点“t2”的所有路径，使用文本相似度函数计算食谱属性的“t1”和“t2”的相似度为排序依据进行排序，将用户的交互记录中的偏好在知识图谱上食谱主食材的属性上进行了两跳的传播，以挖掘用户潜在偏好的食谱。

## 5.5 本章小结

本章介绍了基于知识图谱推荐模型 RippleNet，并介绍了算法的框架和具体实现。首先介绍了本项目的食谱数据集的获取和构造过程，并对数据集进行预处理，其次介绍了数据增强算法，并通过实验确定最佳 AUC 的模型超参数，同时也指出模型推荐准确率较低的原因。

## 第 6 章 系统的详细设计与实现

本章将总体设计作为基础，对系统各个模块进行设计和实现。首先进行配置前后端系统的开发环境，其次对登录注册模块、食谱浏览模块、社区交流模块、个人中心模块等内容进行展开设计与实现，其中使用 SpringBoot、Mybatis、Vue.js、Node.js 等技术框架，以 MySQL 数据库做数据库持久化操作，以 Redis 作为缓存进行实现。

### 6.1 开发环境搭建

本系统采用前后端分离开发模式，在系统的开发过程中，前端独立开发，互不干扰。前端负责发送请求和获取后端返回的数据，并将数据渲染到页面上；后端负责向前端暴露接口，处理相关的业务逻辑，对数据库进行基本操作，完成数据的持久化。所以，本项目在环境的搭建中包含后端环境的搭建和前端环境的搭建。

#### 6.1.1 前端环境搭建

##### 6.1.1.1 前端项目的创建

首先在本地安装 Node.js、vite 工具、Visual Studio Code 编辑器和代码管理工具 Git。其中，Node.js 为 vue 的 JavaScript 代码提供所需的运行环境，vite 是用于构建 vue 项目、配置 vue 项目的工具，提供了多款插件，易于项目的扩展。Git 是一款代码管理工具，可以将系统代码实时进行上传和管理，方便本地代码的扩展和回退，保证系统开发阶段的稳定性和安全性。

在进行前端项目的环境搭建时，首先需要从 Node.js 官网上将 Node.js 进行下载和安装，并按照需求配置相关的环境变量。在下载安装完成后，为方便后续扩展包和插件的便捷下载和管理，需要使用 npm 命令下载镜像源命令 cnpm。在上述步骤完成后，打开 Visual Studio Code 创建一个工作目录，在该工作目录的控制台中使用 vite 工具命令（npm init vue@latest）构建项目，项目名称为 ystj-web。在构建完成项目后，需要使用命令（cnpm install）进行安装项目启动需要的依赖库。在上述所有准备工作完成后，可以通过命令（npm run dev）启动项目，点击生成的 URL 地址，验证项目是否能够正确运行。

前端项目包含许多必要的目录结构，每个目录结构下内容均有特定要求。文件夹 `node_modules` 存放了项目的依赖文件，包含很多基础的依赖库和自定义安装的依赖，往往不需要开发人员手动修改，只需使用 `cnpm` 或 `npm` 命令即可进行添加依赖文件。`public` 文件夹主要存放一些公共资源，如 `logo`、公共的 `CSS` 文件等内容。项目的核心文件存放在 `src` 中，包含项目的源码和各种静态资源，是项目开发过程中的重点目录。在后续的开发过程中可以根据需求适时调整和引入不同文件夹，并丰富其中的内容。特别的，`main.js` 是浏览器最先解析的入口文件，在其中可以引入全局变量和全局组件，如后端 URL 地址变量，Element-Plus 组件等。

### 6.1.1.2 前端项目的配置

使用 `cnpm` 命令导入 `Axios`、`router`、Element-Plus 依赖，以便后续开发使用。由于浏览器的同源策略，前端和后端必须通过配置跨域才可实现信息的交流与传递。`vite` 提供了较为简单的跨域配置，只需配置根目录下的 `vite.config.js` 中 `proxy` 对象，设置跨域前缀为“/api”，配置前端和后端地址和端口号。

项目需要配置响应拦截器，响应拦截器能够根据请求返回的响应状态码来执行拦截器的内容。其中，使用 `ElMessage` 来显示返回的描述。为方便后续便捷使用 `Axios`，本项目依照 `Axios` 请求的构造方式，分别封装了 `post`、`put`、`get` 和 `delete` 请求，请求参数为 `url` 和 `params`。

## 6.1.2 后端环境搭建

### 6.1.2.1 后端项目的创建

后端项目使用的开发工具有 `JDK1.8`、`Maven`、`Git`、`IDEA`，使用的框架主要有 `SpringBoot`、`SpringSecurity`、`JWT`、`Mybatis-Plus` 等。`JDK1.8` 是 Java 开发的工具包，`IDEA` 软件提供了完整的 Java 开发集成环境，具有良好的代码提示功能。`Maven` 是管理 Java 依赖包的工具，只需简短的几行 `XML` 语句便可导入相关的依赖内容。其中，`JWT` 即 `JSON Web 令牌`，用于在通信双方之间安全地传递信息<sup>[19]</sup>。

在后端项目创建前，需要在本地下载安装 `JDK1.8`、`Maven`、`Git` 和 `IDEA`，其次在 `IDEA` 中配置 `JDK1.8` 和 `Maven` 的路径，并将 `Maven` 的本地库路径从 C 盘更改到非系统盘（D 盘）。在配置完成后，使用 `IDEA` 软件新建 `SpringBoot` 项目，命名为 `ystj-server`，并将各个框架对应的依赖语句写入 `pom.xml` 中。后端项目主要包含以下几种文件夹：`controller` 包主要存放向前端提供接口，接收请求并进行业务逻辑处理

的类； service 包主要存放了处理具体的业务逻辑的实现接口，接口的实现文件存放 在该包下的 impl 包中； pojo 包主要存放与各个数据库表所对应的实体对象； mapper 包主要存储了 mapper 接口，每个接口的方法名都与相应的 xml 配置文件的 id 相同， 编写的 SQL 语句都存放在该 xml 配置文件中。除了这些包之外，在不影响项目结构 的情况下，可以以功能类型为单位划分包，如配置包（config）、工具包（utils）等。

### 6.1.2.2 后端项目的配置

当后端项目创建完成后，在项目中导入 Swagger2 等项目所需依赖，并在 config 包下对 SpringSecurity、Swagger2、JWT 令牌工具、Redis 序列化、验证码生成、静态 资源映射（访问后端图片）等内容进行配置，并依据类所对应的属性赋予不同的注解 （@Configuration、@Component 等）。其中@Configuration 注解用于标识配置类， @Component 注解是主要用于在项目中实现组件的自动扫描和依赖注入。配置 SpringSecurity 框架涉及密码加密、认证过滤器、访问接口异常的自定义返回结构等。

为了统一接口的返回数据类型，本项目定义了公共返回类，包含多个返回结果的 静态成员函数，其成员变量和成员函数如表 6-1 所示。

表 6-1 RespBean 公共返回类

类型或返回参数	名称	参数	描述
long	code	—	返回代码。200：成功 500：失败
String	message	—	返回描述，如“验证码错误”等
Object	data	—	返回数据
boolean	state	—	是否将描述显示（ElMessage）
static RespBean	success	message, obj, state	返回成功信息
static RespBean	error	message,obj,state	返回失败信息
static RespBean	success	message,state	返回成功状态
static RespBean	error	message,state	返回失败状态

### 6.1.2.3 查询分页的配置

由于食谱和其他信息众多，免不了出现分页的情况，所以此处先封装好分页的实 体类。本项目主要采用 PageHelper 插件进行分页，PageHelper 的主要原理是在 Spring Boot 启动时，为 SqlSessionFactory 添加了 PageInterceptor 拦截器，这个拦截器会在 SQL 语句执行之前添加分页信息（LIMIT）。如表 6-2 所示，为分页插件的封装实体 类的结构组成。

表 6-2 QueryPageParam 分页实体类

属性	变量名	描述
Integer	pageSize	分页大小（默认为 10）
Integer	pageNum	页号（默认为 1）
HashMap<String, Object>	param	接收前端传来的其他参数

### 6.1.3 数据库搭建和配置

本项目中数据的持久化操作主要使用 MySQL 数据库实现。在保存食谱的点赞数据、收藏数据、浏览数据等内容时，本项目使用 Redis 缓存进行存储，并设置定时任务及时将 Redis 缓存的信息回写到 MySQL 数据库中。Redis 主要用于保存本系统频繁访问的数据，以提高访存效率。

#### 6.1.3.1 MySQL 配置

将 MySQL 的配置内容详情写入 SpringBoot 的配置文件（application.yml）中，配置内容包含 MySQL 数据源的 URL、主机端口、账号、密码等信息，并设置时区。根据上述数据库表设计内容，编写创建表的 SQL 语句，完成数据库创建工作。此外，将食谱数据按照食谱表（t\_recipe）的字段要求处理食谱数据（Excel），再将处理后的食谱数据导入数据库中。

#### 6.1.3.2 Redis 配置

使用 VMware 创建虚拟机，下载并安装 CentOS 系统，将 Redis 安装包使用 MobaXterm 软件导入虚拟机中，然后使用命令 tar 进行解压，完成 Redis 安装工作。完成 Redis 的安装流程后，需要对 Redis 的配置文件修改，Redis 的端口号修改为 8090，取消保护模式。在 SpringBoot 的配置文件中写入 redis 的 ip 地址和端口号。特别的，编写 Redis 序列化文件后放入 config 包下，并以@Configuration 和@Bean 进行标识，以便后续使用注解@Autowired 便可直接使用 RedisTemplate。为了操作 Redis 更加方便和简洁，后端项目对 RedisTemplate 根据具体的业务需求设计了一系列的操作接口，例如设置键值、删除键值等，这样无需时常关注具体 RedisTemplate 的实现细节。

在本项目中，系统使用 Redis 缓存数据库对食谱的点赞、浏览、收藏等数据进行缓存，以缓解对 MySQL 数据库频繁访存的压力。项目设计实施的过程中，使用工具 RedisDesktopManager 对 Redis 数据库进行调试和管理。项目在利用 Redis 存储数据时，会根据存储内容的种类编写 key 的构造表。

## 6.2 登录注册详细设计与实现

登录注册模块是用户端系统和后台管理系统中非常重要的功能模块，该模块主要向用户提供接入本系统的接口。虽然，未登录用户可以登录本系统查看一些食谱详情，但是不能使用点赞、收藏、获取推荐等功能。在后端系统中，该模块还承担了依据用户角色动态生成菜单的功能。由于用户端和管理员端的系统登录注册模块基本一致，所以本文只在此处详细说明该模块的详细设计和具体实现方法。

### 6.2.1 登录注册的流程设计

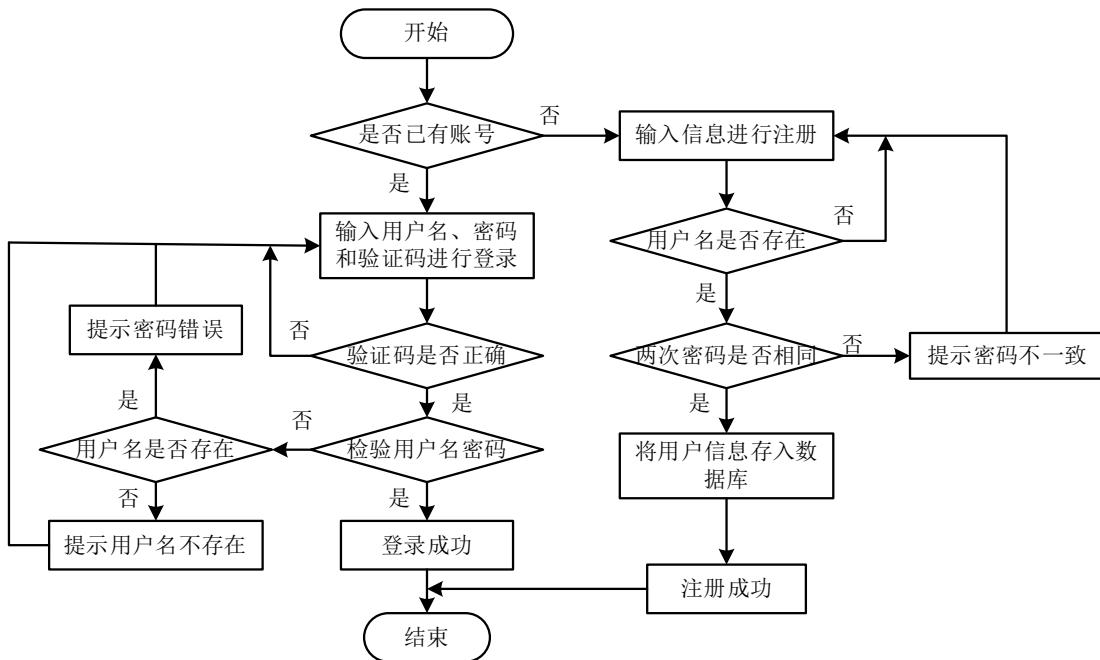


图 6-1 登录注册流程图

如图 6-1 所示，为系统的登录注册流程图。当用户已有账号时，则可以通过登录页面进行登录。前端将用户名、密码和验证码进行封装，通过请求发送给后端，后端则依次校验验证码、用户名和密码是否正确，如果不正确则向前端返回必要的提示信息以引导用户下一步的操作。当用户没有账号时，则可以通过注册页面进行注册，前端将通过表单验证的必要信息进行封装，并给后端发送注册请求，后端则要先从数据库中判断用户名是否存在。如果不存在，则将用户信息进行保存到数据库，注册成功；如果存在，则返回必要的提示信息。特别的，当管理员在登录后端管理系统时，会在登录成功后向后端发送请求以获取到该管理员所属角色可见的菜单信息，并将该菜单信息通过 Element 控件动态渲染到前端页面，使得管理员仅可见具有权限的

菜单信息，从而进行后续的操作管理工作。

用户在前端输入用户名、密码等信息时，前端需要按照提前设定的规则（密码长度在 6-20 位之间等）进行校验表单信息。

### 6.2.2 登录注册的类设计和实现

系统的登录注册模块主要涉及到的类有控制器 LoginController、控制器 CaptchaController、用户接口类 UserService、用户实现类 UserServiceImpl、数据库接口类 UserMapper、实体类 User、实体类 UserLoginParam 和其他工具类。

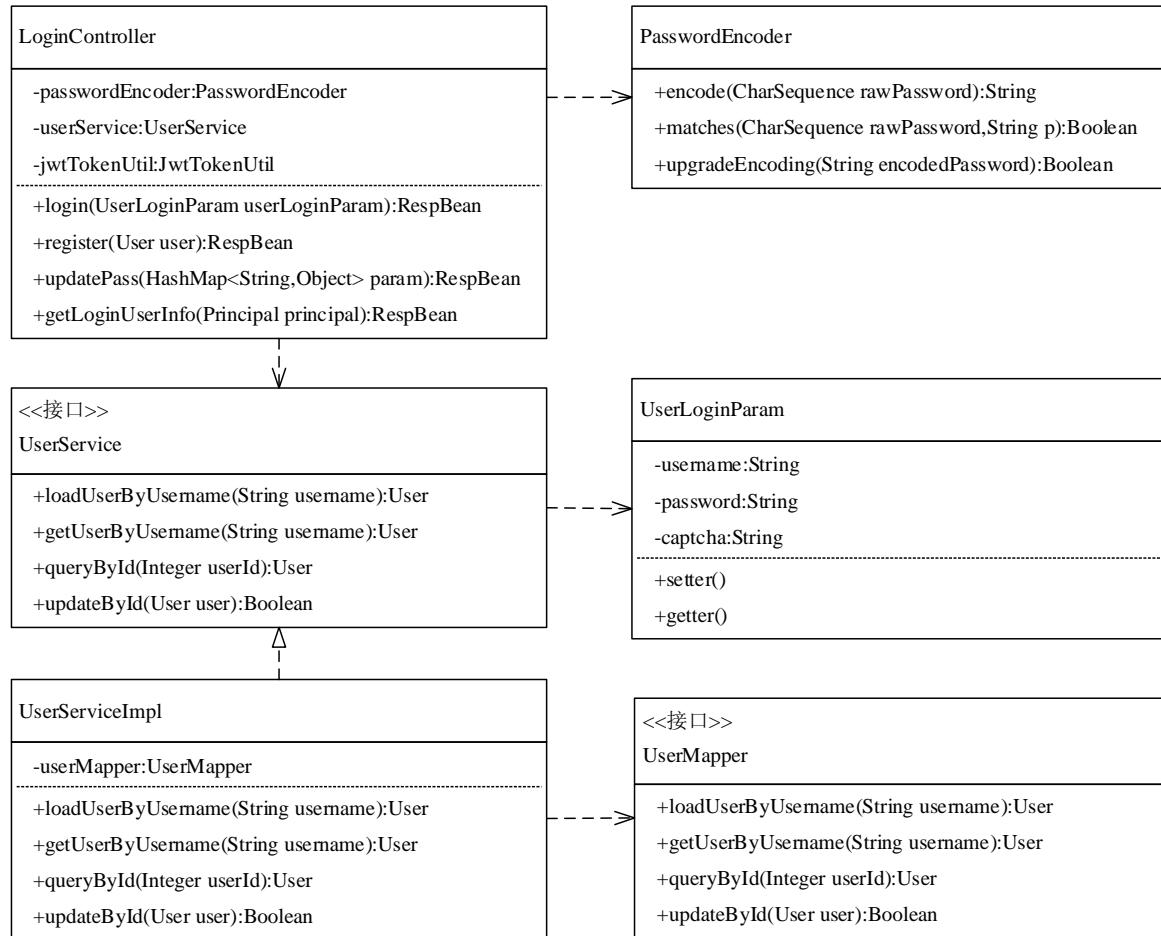


图 6-2 登录注册模块类图

如图 6-2 所示，为登录注册模块类图设计，其中 UserServiceImpl 类实现了 UserService 接口，UserServiceImpl 使用 PasswordEncoder 类完成对用户密码的加密、解密和匹配工作，以确保用户的个人信息安全。

如表 6-3 所示，为 LoginController 接口表，主要向当前登录用户提供一些业务逻辑服务，其中涉及到的实体类有 UserLoginParam，其中包含用户名、密码和验证码三

个变量。这个实体类仅在用户登录使用，可以防止滥用 User 类和其中包含的变量，防止冗余和程序错误的发生，简化了前端登录请求参数的构造流程。

在该控制器中，主要包含登录返回 token、注册用户信息和获取当前用户信息的功能。特别的，在返回当前登录用户信息的接口参数是 Principal 类型，它是 Spring Security 通过验证后自动提供的变量，无需前端进行构造返回，通过 Principal 即可获取登录时保存的用户名，进而得到当前的登录用户信息。

登录注册模块是用户端系统和管理员端系统都具有模块，其后台使用的数据库表是相同的。

表 6-3 登录注册模块接口表

URL 地址	参数类型	描述
POST /login	UserLoginParam p	登录，如果登录成功，则返回 token
POST /register	User user	注册用户信息，注册成功，则返回 token
GET /getuserinfo	Principal p	返回当前登录用户信息
POST /api/captcha	—	获取验证码图片

#### 6.2.2.1 登录功能

当用户进入到登录页面时，前端登录页面的验证码图片标签中设置的 src 为 “’/api/captcha?time=’+new Date()’”，由于存在 “/api” 前缀，则前端会向后端发送请求获取验证码，其中请求 URL 中 time 参数是为了能够区别每一次的发送请求，以防止多次刷新由于浏览器的缓存机制无法显示新验证码的情况。当用户按照要求输入登录的用户名、密码和验证码时，el-form 表单会根据提前设置的 rules 规则，判断输入的表单信息是否符合要求，并使用 form 对象存储用户填写的账号、密码和验证码。用户点击登录按钮触发函数 loginForm()，该函数会再次验证 form 表单数据是否符合提前预定的规则，如果符合，使用 Axios 向 “/api/login” 发送 post 请求。

后端成功接收到请求后，首先，从 HttpServletRequest request 对象中取出请求到的验证码信息，然后验证输入的验证码信息是否正确，如果验证码不正确则直接返回 RespBean.error，提示信息为“验证码输入错误，请重新输入”；如果验证码正确，则通过 username 从数据库中获取是否有该用户信息，查询出该用户信息后通过 passwordEncoder 校验输入密码是否正确。如果正确，则返回 HashMap 对象，其中包含字段 “token” 用于保存 JWT 登录令牌，字段 “tokenHead” 用于标识令牌头；如果不正确则返回必要的提示信息。

如果登录成功，前端接收到返回的 token，使用 window.sessionStorage 将 token 令牌保存在浏览器本地存储中，并通过 \$router 跳转到“/main”主页中。

### 6.2.2.2 注册功能

注册功能的前端判断和操作与登录功能类似，此处不再赘述。在注册页面，点击注册按钮后触发 register 函数，该函数使用 Axios 向“/api/register”发送 post 请求，请求参数为 form 对象。后端成功接收到请求后，先依次判断验证码是否正确，用户名是否存在，如果验证码正确，且用户名存在，则将注册的信息通过 userService 对象保存到数据库中，并向前端返回含有用户信息的令牌。前端接收到请求的成功响应后，将 JWT 令牌保存到浏览器本地存储中，然后通过 \$router 跳转到“/main”主页中。

在用户信息方面，密码内容必须做到加密，以提高系统的安全性和稳定性。本项目在获取到用户注册的信息时，会使用 Spring Security PasswordEncoder 进行密码加密，再将加密后的密码存储到数据库中。用户登陆也是用该类提供的接口进行匹配。特别的，查询用户信息时，密码变量被设置为 NULL 再进行返回。

### 6.2.2.3 接口访问权限

后端需要限制未登录用户可以访问哪些接口，而登录用户可以访问哪些接口，Spring Security 通过配置权限忽略项进行配置开放的接口。对于某些特定场景，如食谱界面，未登录用户可以进行查看；当未登录用户执行点赞操作时，则提醒用户未登录，并跳转到登录页面。

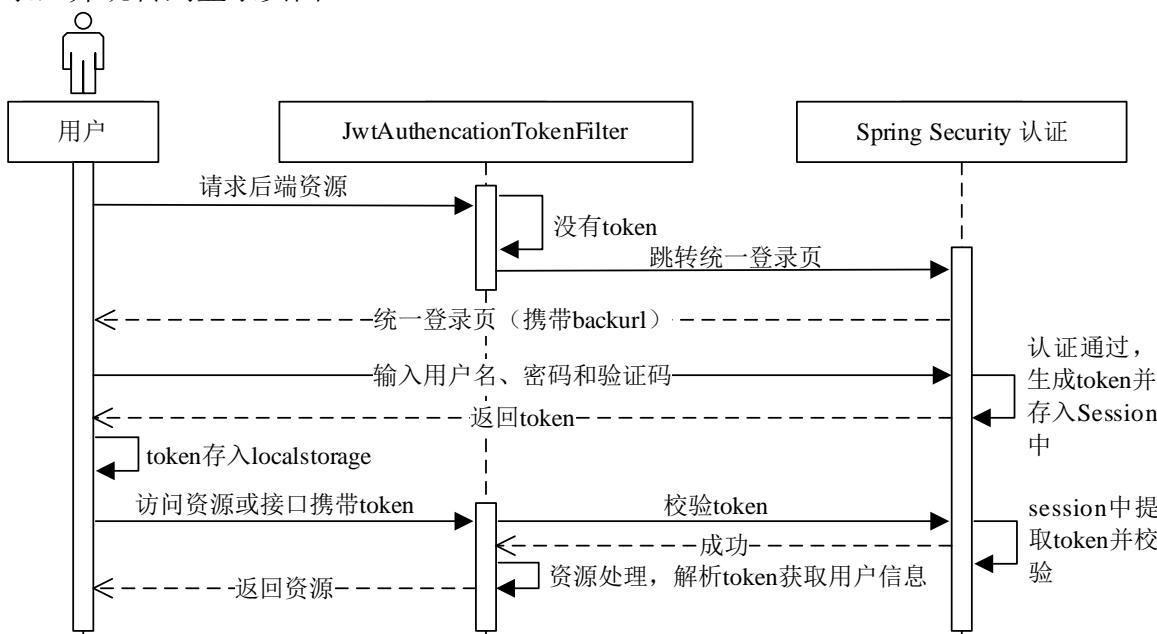


图 6-3 系统请求时序图

如图 6-3 所示, 为系统请求时序图。当用户访问需要登录的后端资源时, 过滤器会检测到请求中没有 token, 则通过认证中心返回统一的登录页面的提示信息。用户需要进行登录, 当用户认证成功后, 将生成的 token 存入 Session 中, 并返回 token。前端会将 token 存入到 localstorage 中。前端的过滤器会在每次发送请求时, 将 token 封装到请求中。当用户再次发送请求时, 会从 session 中提取 token 并进行校验。

## 6.3 用户端系统详细设计与实现

### 6.3.1 食谱浏览模块

食谱浏览模块主要完成向用户进行展示推荐食谱结果、食谱详情内容如发布人用户、食谱详情内容、点赞食谱和收藏食谱, 也可以通过关键词搜索或类别进行筛选食谱。在食谱详情页中, 用户可以点击发布人的信息跳转到该发布人的主页中, 并查看该用户所发布、点赞和收藏的食谱信息, 也可以查看该用户参与的讨论。

#### 6.3.1.1 食谱浏览模块流程设计

(1) 食谱列表浏览 如图 6-4 所示, 为食谱列表浏览流程图。在食谱列表浏览中, 共包含多个食谱小组件。每个小组件包含食谱的信息, 发布人的信息等内容。

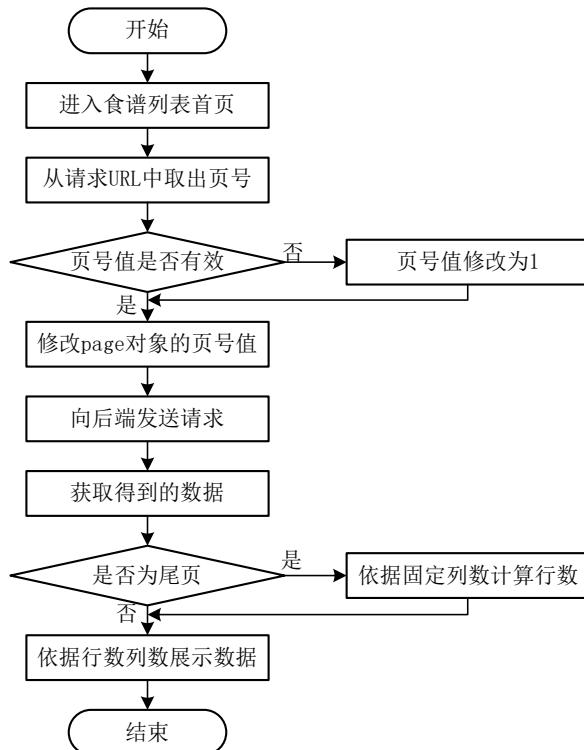


图 6-4 食谱列表浏览流程图

进入到食谱列表展示页面后，首先从 URL 中取得页面号，判断该页面号是否有效；如果有效则修改 page 对象中的页面号；如果无效则将修改页面号的值为 1，也一并修改 page 对象的页面号。其次，向后端发送请求获取该页面号所需展示的食谱数据。由于返回数据是由 PageInfo 封装的对象，其中包含页面总数、当前页大小、食谱列表等信息。通过该对象的值可以优先判断当前页号是否为尾页，如果是尾页，则需要依据固定的列数计算出行数；如果不是尾页，则直接进行渲染数据即可。这样可以避免在循环渲染数据时，出现得到空值的情况，从而导致程序错误。

**(2) 食谱详情浏览** 如图 6-5 所示，为获取食谱详情流程图。当用户点击进入食谱详情页时，会携带参数进行跳转到食谱详情页。首先，解析 URL 中 id 的值，并将其封装后向后端发送请求以获得该 id 下的食谱详情信息。在后端中，需要从数据库中查询到该食谱的信息，然后从 Redis 中获取是否有该食谱的浏览数，如果没有则浏览数自增加 1 存入 Redis 中，如果有则将浏览数自增加 1 并取出。其次，以同样的方式找寻 Redis 中是否有该食谱的点赞和收藏数据。最后，判断用户的 token 是否有效，如果有效则在该食谱的点赞和收藏列表判断该用户是否点赞和收藏，如果是则将用户对象中是否点赞和是否收藏变量设置为 true，并将食谱返回到前端进行显示。

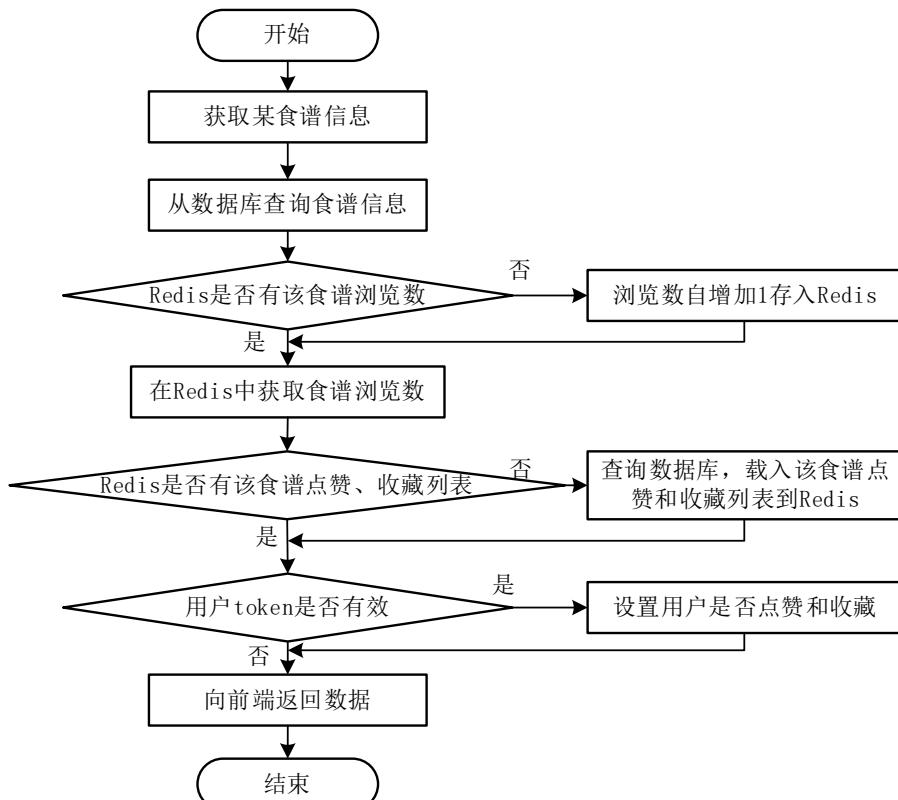


图 6-5 获取食谱详情流程图

(3) 食谱点赞和收藏     如图 6-6 所示, 为食谱点赞的流程图。当用户点击进入食谱详情页时, 会在 vue 页面数据挂载时的生命周期中 (mounted) 获取到食谱的数据。其中食谱的数据中也包含用户的一些字段信息, 如用户的 id、用户是否点赞该食谱等内容。vue 页面会根据食谱的具体数据来渲染组件, 例如用户已经点赞了该食谱, 则点赞图标为“已点赞”状态, 其他收藏功能也是如此。在食谱点赞流程图中, 用户再次点击“已点赞”状态的点赞按钮时, 前端页面会弹出 ElMessage 来提示用户当前已经点赞。如果用户还未点赞, 则前端会向后端发送点赞请求, 请求参数为食谱 id, 并由拦截器自动添加本地的用户 token 以唯一标识登录用户。后端在接收到点赞请求后, 会先判断 Redis 当中是否存在该食谱 id 的点赞列表和点赞数量的 key, 如果不存在, 则构造相应的 key 存储到 Redis 中, 以便后续的操作。其次, 后端过滤器会验证用户的 token 是否有效, 如果无效则返回信息让用户跳转到登录页面。如果用户 token 有效, 则通过 token 解析出用户 id, 并将其加入到该食谱的用户点赞列表中, 最后对该食谱的点赞数量自增加 1。特别的, 在后端系统中, 对于类似存储到 Redis 中的数据均会设置定时任务, 将 Redis 中数据定期向 MySQL 中进行存储。

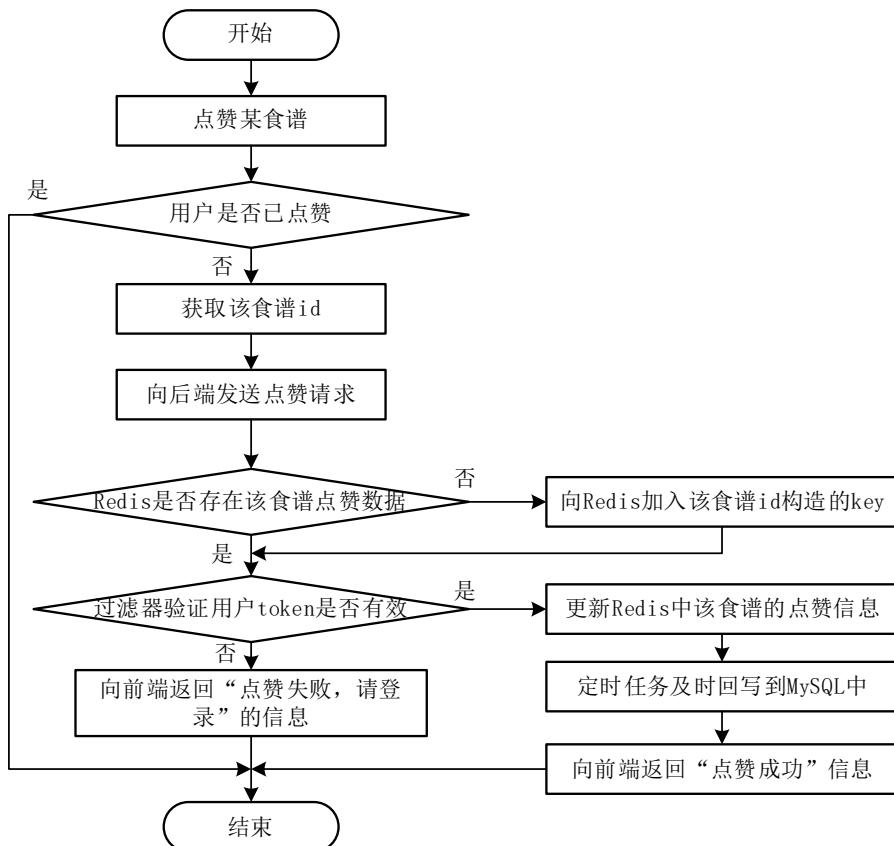


图 6-6 食谱点赞流程图

### 6.3.1.2 食谱浏览模块类设计与实现

(1) 后端实现 食谱浏览模块设计到的主要类有控制器 RecipeController、控制器 RecipeOpController、食谱接口类 RecipeService、食谱接口实现类 RecipeServiceImpl、食谱操作接口类 RecipeOpService、食谱操作实现类 RecipeOpServiceImpl、食谱实体类 Recipe、用户实体类 User、食谱操作实体类 RecipeOp 和其他工具类。如图 6-7 所示，为食谱浏览部分类设计。

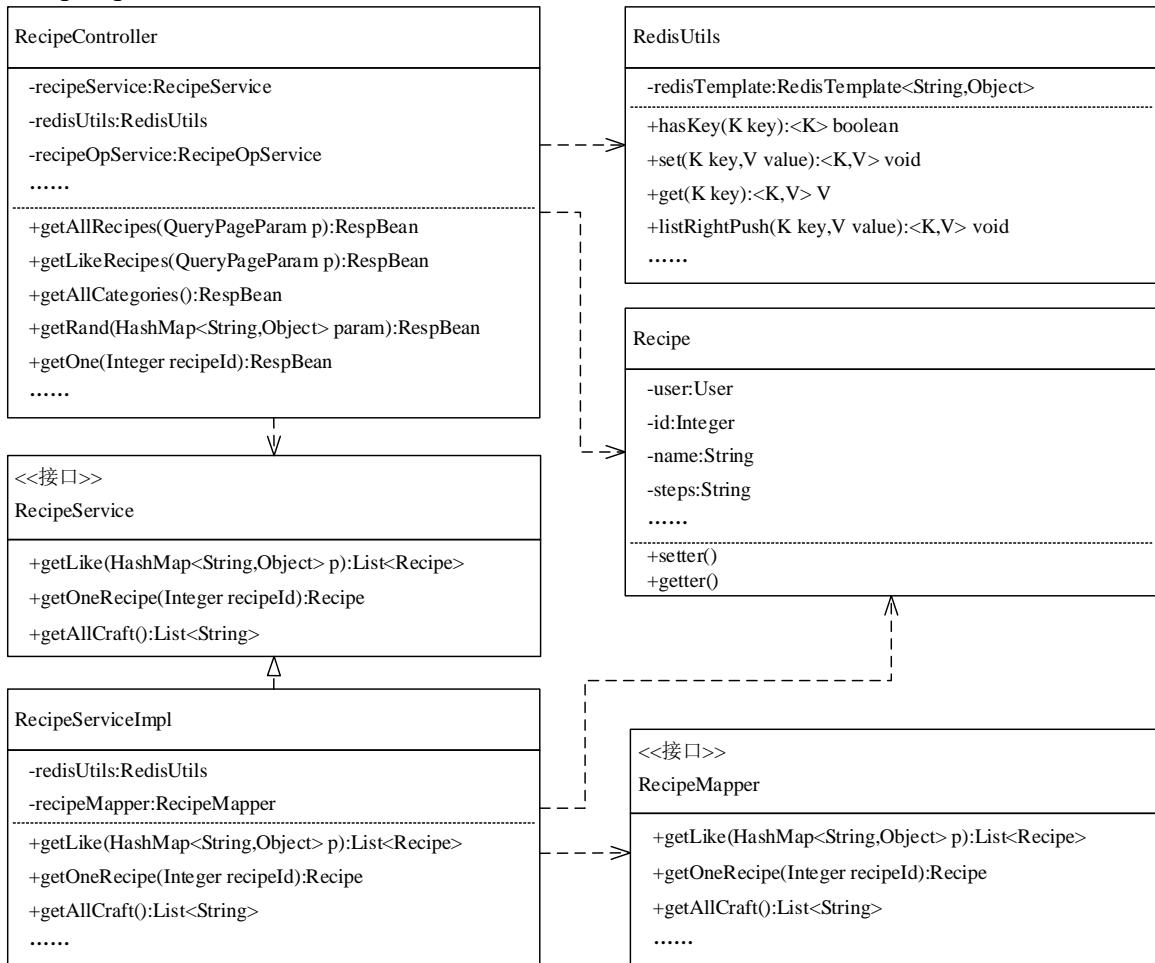


图 6-7 食谱浏览类图设计

RecipeController 用于向前端暴露接口，为前端提供服务。当 RecipeController 接收到前端发送的请求后，RecipeController 会根据需要处理的业务逻辑来调用 RecipeService 类中的方法。RecipeServiceImpl 是 RecipeService 的实现类，用于调用 RecipeMapper 中 SQL 语句，并返回封装的结果。Recipe 类中包含了 user 对象，它用于存储该食谱的发布用户，利于前端显示食谱的发布人头像和昵称，也利于跳转到该发布人用户的主页。要实现查询结果中包含 User 对象，首先，需要在 RecipeMapper.xml 编写的 SQL 语句中，使用 LEFT JOIN 将用户表中的数据左连接到

食谱表中，并根据所需的筛选条件编写 where、order 等其他内容，为 select 关键字中结果编写别名；其次，使用 resultMap 标签编写查询结果映射 MyRecipesResult，其中使用 association 标签指明 Recipe 类中所需要映射的对象 user，并在其中标识 select 查询结果中 user 的别名与对象 user 的变量对应关系；最后，修改 select 标签的 resultMap 属性为自定义的映射结果 MyRecipesResult。

在 Controller 向 Service 传递参数时，通常使用 HashMap 类型进行传递。这样 Mapper.xml 文件能够配置 paramType 属性为 java.util.Map，可以使用#{key} 根据 HashMap 的 key 来访问到 key 所对应的 value 值。

如表 6-4 所示，为本模块中所使用的接口表。

表 6-4 食谱浏览模块接口表

URL 地址	参数类型	描述
POST /recipe/getAll	QueryPageParam p	查询所有食谱信息（分页）
POST /recipe/getLike	QueryPageParam p	模糊查询食谱信息（动态 SQL）
GET /recipe/{id}	String id	查询 id 为某值的食谱信息
GET /getAllCategories	—	获取所有的类别（工艺、口味等）
PUT /recipeop/star/{id}	String id,Principal p	登录用户点赞 id 为某值的食谱
PUT /recipeop/nostar/{id}	String id,Principal p	登录用户取消点赞 id 为某值的食谱
PUT /recipeop/collect/{id}	String id,Principal p	登录用户收藏 id 为某值的食谱
PUT /recipeop/nocollect/{id}	String id,Principal p	登录用户取消收藏 id 为某值的食谱

(2) 前端实现 在本模块中，前端页面依旧遵循模块化开发的原则。食谱列表页面文件为 Recipes.vue，食谱列表中每个食谱的小卡片为 OneShowRecipe.vue，Recipes 的 vue 文件中利用 el-row、el-col 和 v-for 循环动态生成 OneShowRecipe 子组件，这样可以构建完整的食谱列表，其中默认为 6 列。Recipes 父组件进行渲染时，会执行 mounted 钩子函数中 initData 函数，该函数用于根据 page 对象中的分页信息，获取食谱的数据。其次，Recipes 父组件只需要利用 props 方式向子组件传递一个具体食谱信息，子组件便可根据获取的食谱信息进行渲染食谱小卡片，进而动态生成食谱列表。当用户在 Recipes 父组件点击切换页面时，会触发 handleCurrent 事件，更改 page 对象中的分页信息，并再次调用 initData 进行更新和渲染数据。特别的，为了避免在更新食谱数据时页面产生短暂白色画面的现象，在最外层的 div 标签属性种加入

v-loading="loading"属性，并在 initData 中将 loading 置为 true，在成功获取到数据后，使用 setTimeOut 设置 300 毫秒的延迟后将 loading 置为 false。这样可以在异步加载数据时，页面能够显示加载动画，当数据获取完成后，再结束加载动画。

用户可以点击小卡片，会触发 enterRecipeDetail 函数，该函数利用\$router.resolve 解析一个路由。通过指定路由的 name 为“RecipeInfo”，并在指定的路由中设置 query 参数为当前小卡片食谱对象的 id，其次再通过 window.open 方法以新窗口（\_blank）的方式打开解析得到的路由链接，这样就可以跳转到某个食谱的详情页，例如其中一个食谱的详情页 URL 为/recipe?id=1001，通过\$router 打开一个新的浏览器窗口，页面的地址为该详情页的 URL。

用户进入详情页面后，前端会载入 RecipeInfo 的 vue 文件。首先，执行 mounted 中的 initData 函数。该函数会从路由参数中获取到菜谱的 id，然后使用 getRequest 函数发送 GET 请求，请求 URL 为 “`/api/recipe/\${id}`” 以此获取指定 id 的菜谱详情信息。其次，通过 Promise 链式调用获取响应对象的数据部分。菜谱数据多个字段的值需要再次处理，包括将菜谱图片信息按照“@”符号分割为数组；如果菜谱的主料或辅料不为空，则将其按空格分割为数组，并计算渲染数据所需布局的行数和列数；如果菜谱的步骤字段不为空，则将其按照“||”分割成数组，每一个元素为一条步骤，并将每条步骤再按照“@”分割为数组，以此分离每条步骤的文字与图片；将菜谱的适合人群字段信息按“，”分割成数组。在数据处理完后，将菜谱数据赋值给 recipe 数据属性，并根据请求返回的数据信息设置其他属性的值，如当前登录用户是否已点赞、是否已收藏等。最后，在 300 毫秒延时后隐藏加载状态，以确保用户体验和交互流畅。食谱详情页面需要渲染上述处理的各种信息，包括工艺、口味、所需时间、难度、主料、辅料、发布用户人信息等。在显示发布用户人信息块中，用户可以点击该信息块触发@click 点击事件，该点击事件使用\$router.resolve 方法指定了路径为“/usercenter”的路由，该路由的 query 参数为用户的 id，以此进入用户的个人主页，查看该用户的操作过的食谱和参与交流的讨论。

### 6.3.2 社区交流模块

社区交流模块是本系统的另一主要模块，它为注册到系统的用户提供了一个交流平台，旨在促进用户之间进行交流和分享。用户可以对他们喜欢的主贴进行评论，也可以分享他们自己的食谱和心得，亦或是提出一些关于食谱制作、烹饪技巧等方面

的问题。社区交流模块会提供热门和最新内容的展示功能，让用户可以快速浏览到最受欢迎和最新发布的食谱、评论和问题，从而及时了解社区动态。

### 6.3.2.1 社区交流模块流程设计

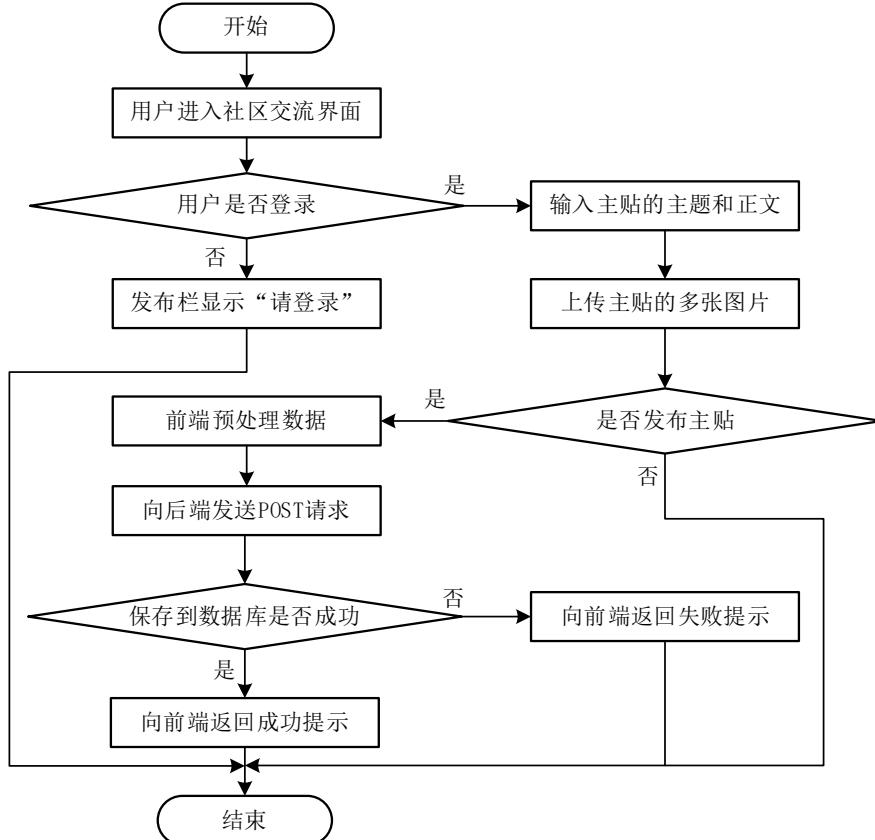


图 6-8 发布主贴流程图

如图 6-8 所示，为发布主贴流程图。用户进入社区交流界面后，可以根据当前的热度等方式进行排序，查看其它用户发布的主贴列表，在主贴列表中点击可以进入某个主贴详情页，并查看该主贴所包含的其它用户的回复。此外，无论是主贴列表还是主贴详情页都包含发布用户的头像、用户名以及发布时间，只需点击发布用户的信息便可进入该用户的个人主页。用户进入社区交流界面时，前端会判断本地的 `localStorage` 是否有 `tokenStr` 所包含的值，如果没有则说明用户未登录。未登录用户无法发布主贴，发布主贴组件显示“用户未登录”。登录用户可以在发布主贴组件中输入主贴的主题和正文内容，并可以上传多张图片。在用户完成相关内容的输入和上传后，用户可以点击发布主贴按钮。前端会将相关内容进行处理，如将图片的 URL 列表转化为以“@”符号分隔的字符串。其次，将处理后的主贴内容通过 `post` 请求发送到后端。后端接收到数据后，将主贴内容加入登录用户的 `id`，并通过 `service` 进

行保存，如果保存成功，则向前端返回成功提示；如果保存失败则返回失败提示。

回复主贴内容也是类似的操作，输入相关的回复内容，并在前端将回复内容和要回复的主贴 id 进行封装，其次将封装好的数据通过 post 请求发送到后端。后端接收到封装好的数据后，将其添加登录用户的 id，然后通过 service 保存到数据库中，如果保存失败则返回相应的失败提示信息；如果保存成功则返回成功提示信息，并重新刷新当前页面数据。

### 6.3.2.2 社区交流模块类设计与实现

如图 6-9 所示，为发布主贴功能类图。该功能涉及主要类有控制器 TopicController、主贴接口类 TopicService、主贴接口实现类 TopicServiceImpl、主贴类型接口实现类 TopicTypeServiceImpl、用户接口实现类 UserServiceImpl、数据库接口类 TopicMapper、主贴实体类 Topic。TopicController 主贴控制器主要包含添加主贴、模糊查询主贴、获取某个主贴详细信息等内容。特别的，TopicTypeServiceImpl 主要负责根据 type 的 id 查询到特定的主贴类型名称，以此返回到前端进行显示，无需二次发送请求以获取主贴的类型名称。

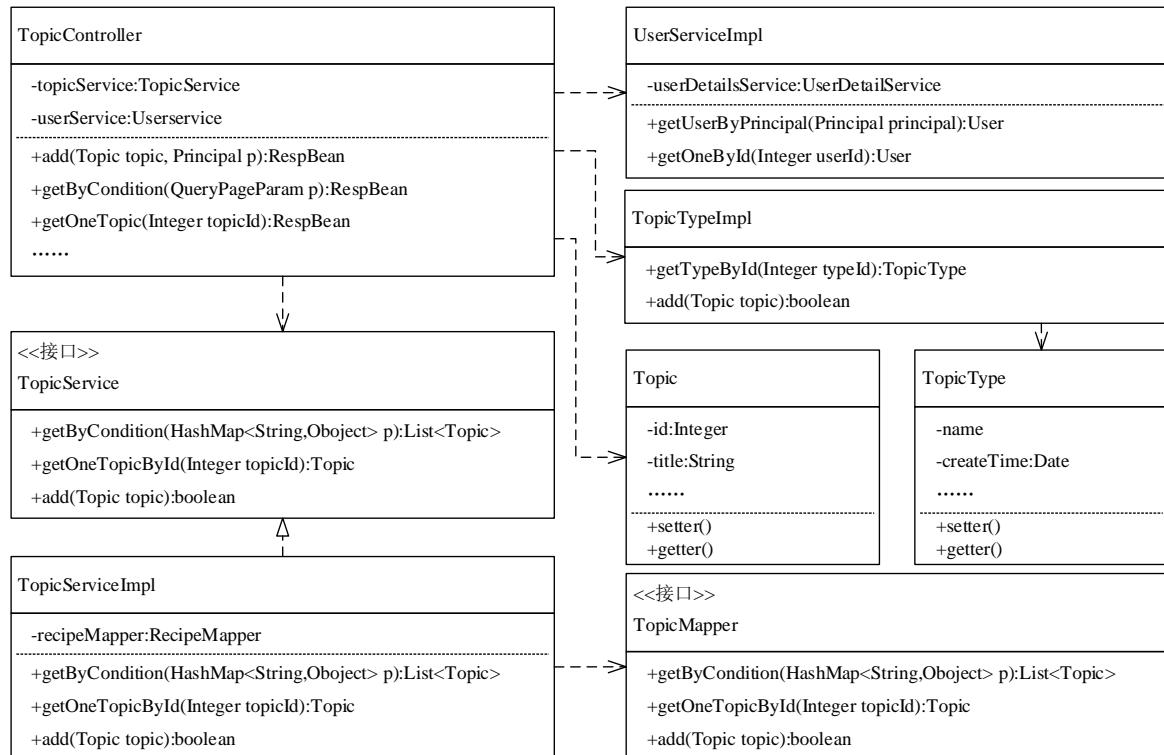


图 6-9 发布主贴功能类图

如图 6-10，所示为社区交流模块回复主贴类图设计。该功能涉及类有控制器 TopicReplyController、主贴回复接口类 TopicReplyService、主贴接口实现类

TopicReplyServiceImpl、主贴类型接口实现类 TopicTypeServiceImpl、用户接口实现类 UserServiceImpl、数据库接口类 TopicReplyMapper、主贴回复实体类 TopicReply、主贴类型实体类 TopicType。主贴控制器中主要包含添加主贴的一级评论回复，添加主贴回复的二级评论，并可以查询到某个主贴的一级评论和二级评论。TopicReply 实体类中含有 childs 变量，该变量用于存放评论的子评论，这样在进入主贴详情页后，便可以直接渲染该主贴评论的二级评论。该实体还存在 publishUserId 变量和 user 变量，便于前端渲染页面时加载发布人的头像、昵称等基本信息。

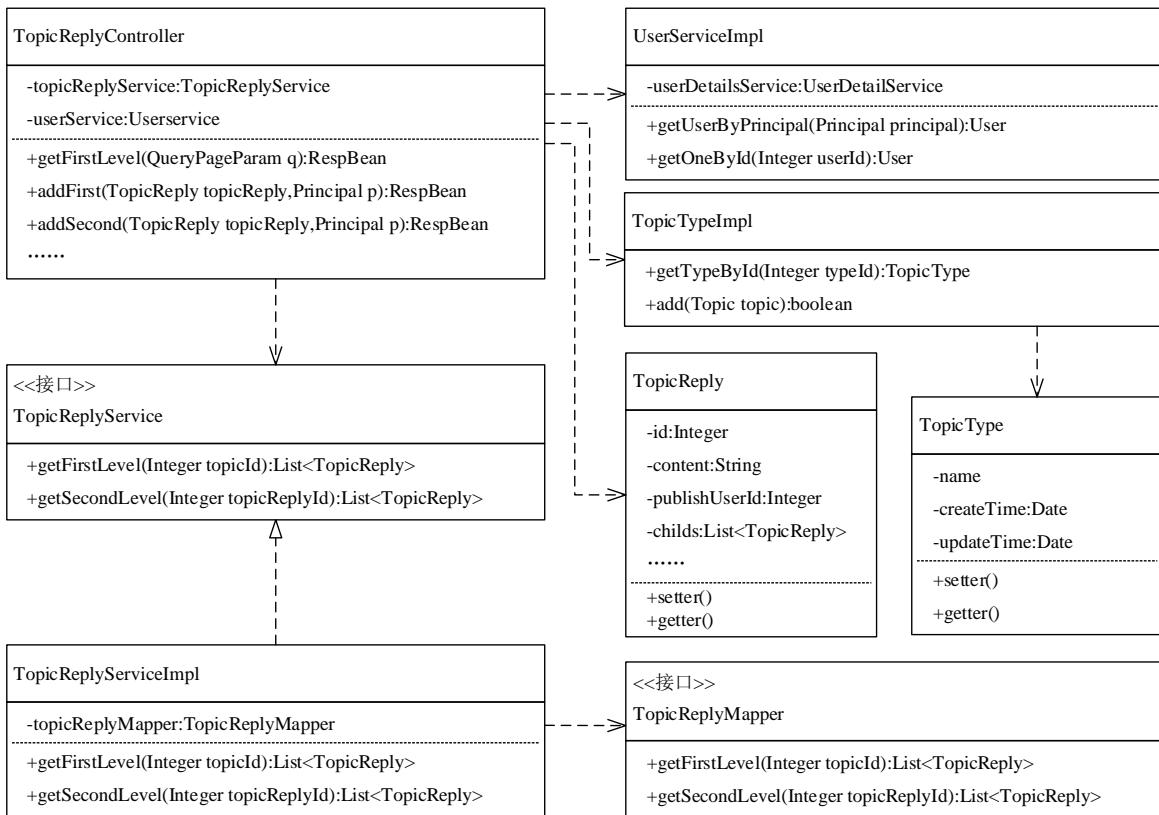


图 6-10 回复主贴类图

本系统类图的设计一般包含 Controller 层、Service 层、Mapper 层，对于其它一些数据处理的方法可能还涉及其它的工具类，类图设计思路大体相近，后续在模块详细设计时由于篇幅有限将不再列举类图，只给出具体的类名。

### 6.3.3 个人中心模块

登录用户可以进入个人中心模块。在个人中心模块，用户可以通过点击修改个人资料，在弹出的对话框中编辑个人信息并进行保存。用户也可以点击头像选择图片并进行裁剪后成功修改个人头像，也可以对登录密码进行修改。此外，登录用户可以在

个人中心查看自己已点赞、已收藏、发布的菜谱和发布的讨论。登录用户可以对已发布的菜谱进行删除。个人中心模块可进一步扩展为用户中心，用户可以点击进行查看其它用户的个人中心主页，并且查看该用户的点赞、收藏的食谱等信息。

个人中心模块涉及的主要类有控制器 UserController、食谱控制器 RecipeController、主贴控制器 TopicController、用户接口类 UserService、用户接口实现类 UserServiceImpl、食谱接口类 RecipeService、食谱接口实现类 RecipeServiceImpl、主贴接口类 TopicService、主贴接口实现类 TopicServiceImpl、密码工具类 PasswordEncoder。在本模块中，UserController 主要负责对用户信息进行修改和查询，RecipeController 主要负责对用户发布、收藏和点赞的食谱进行查询，并提供删除用户发布的食谱接口。TopicController 主要完成对用户所发布的主贴进行查询。

表 6-5 个人中心模块接口表

URL 地址	参数类型	描述
GET /user/getOneById	Integer userId	获取某个用户信息
GET /user/getuserinfo	Principal	获取当前登录用户信息
POST /user/update	User user	更新用户个人信息
POST /user/updatepass	HashMap<String, String> hm	修改密码
POST /recipe/getUserStarList	QueryPageParam p	获取某用户点赞食谱列表
POST /recipe/getUserCollectList	QueryPageParam p	获取某用户收藏食谱列表
POST /recipe/getUserPublishList	QueryPageParam p	获取某用户发布食谱列表
DELETE /recipe/delete	Integer recipeId	删除 id 为某值的食谱
POST /topic/getByUser	QueryPageParam p	查找某用户发布的主贴

如表 6-5 所示，为个人中心模块所使用的接口表。个人中心模块可以获取某个用户信息、获取当前登录用户信息、更新用户个人信息、修改密码、获取某用户点赞、收藏、发布的食谱列表，删除 id 为某值的食谱，查找某用户发布的主贴，删除 id 为某值的主贴等内容。

### 6.3.3.1 个人主页设计与实现

用户点击进入个人主页后，会渲染 MineCenter 组件，优先触发生命周期中 created 钩子函数，该函数中调用了 initData 方法。initData 方法中先将 loading 加载状态置为 true，使用 getReqst 方法发送 GET 请求，请求 URL 为 “/api/getuserinfo”，用以获

取当前登录用户的个人信息。后端接收到前端发送的请求后，后端通过安全框架封装了 Principal 对象作为参数，利用 principal.getName 方法获取用户名，再根据用户名调用 userService 的方法获取用户对象，并将其封装为 RespBean 对象返回到后端。后端在接收到请求的成功响应后将数据赋值给 user，并通过 vue 的数据双向绑定和三目运算符将用户数据渲染在页面上。

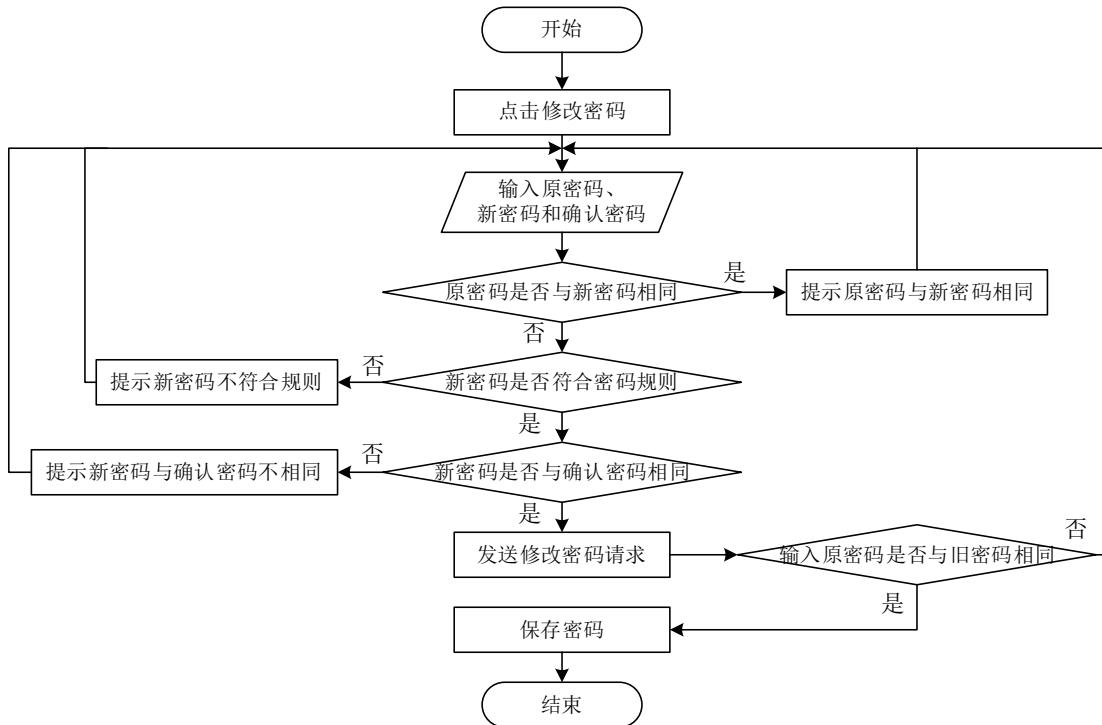


图 6-11 个人中心修改密码流程图

如图 6-11 所示，为个人中心修改密码的流程图。当用户点击按钮进行修改密码时，需要对原密码、新密码、确认密码进行相关的判断，如果不符合条件则会使用 ElMessage 弹出相应的提示信息。

当用户点击修改密码时，会触发函数将 changePassDialog 置为 true，随后弹出密码修改对话框，用户可以填写原密码、新密码和确认密码，并点击修改按钮触发 savePass 函数。该函数会检查输入的两次新密码是否一致，如果不一致则弹出 ElMessage 提示消息；若一致，则检查新密码是否与旧密码相同，如果相同也弹出相应的提示信息；若都通过，且新密码符合密码规则，则使用 postRequest 方法向指定的路径 “/api/user/update” 发送一个包含旧密码和新密码的请求。后端在接收到该请求后，将请求体的数据转化为包含键值对的 HashMap 类型，并接收 Principal 对象。其次，从 HashMap 中获取旧密码 oldPassword 和新密码 newPassword，然后通过

userService 使用 getUserByPrincipal 获取原始用户对象，再使用 passwordEncoder 的 matches 方法匹配输入的旧密码是否与原始密码相匹配，如果不匹配则返回旧密码错误的提示信息。如果匹配，将新密码使用编码器进行编码后设置原始对象的密码字段，最后通过 userService 的 updateById 方法更新用户信息。

用户点击个人头像触发 upload 组件的 on-change 事件，该事件会调用 changeUpload 函数。该函数将表示裁剪对话框的显示状态变量 dialogVueCropper 设置为 true，然后创建一个 FileReader 对象，并通过 reader.onload 事件监听读取完成后，将读取的结果（以 DataURL 形式）给名为 mycrop 的 DOM 元素的 setImg 方法作为参数，来设置裁剪对话框中待裁剪的图片。裁剪对话框内包含封装的 vue-cropper 组件，该组件通过 cnpm 命令进行下载。在裁剪完成后，用户点击修改头像触发 getCrop 函数，该函数通过获取裁剪图片组件 Ref 的 getCropBlob 方法获取裁剪后图片的 Blob 数据，并将该数据利用 FormData 进行封装，再通过 postRequest 方法向“/api/uploadimg”发送 POST 请求。后端在接收到请求后，将类型为 MultipartFile 的图片文件进行重命名、保存到本地等操作，然后将图片的 URL 地址返回到前端。前端在接收到成功的响应后，将图片的 URL 地址赋值给 user 对象的 face 变量中，再通过 saveInfo 函数将用户 user 保存到数据库中。用户进行修改个人资料后，点击保存按钮会触发 saveInfo 函数，该函数使用了 putRequest 方法向“/api/user/update”发送 PUT 请求，请求参数为用户 user。

个人中心的点赞、收藏、发布的菜谱和讨论界面均在 el-tabs 中显示。el-tab-pane 分别包含了 StarRecipes、CollectRecipes、UploadRecipes、UserTopics 四个组件。这些组件的实现方法类似，此处以 StarRecipes 为例进行说明。引入 StarRecipes 组件时，父组件以 :userId 向子组件传值，子组件使用 props 属性接收用户 id。子组件被调用渲染后，会触发生命周期中的钩子函数 mounted，该函数调用了 loadRecipe 方法。该方法中，利用 userId 构造 param 对象，并将该对象赋值给 page 对象的 param 变量中，再通过 postRequest 方法发送 POST 请求，请求的 URL 为“/api/recipe/getUserStarList”，请求参数为 page 对象（包含页面大小、页号和其它参数 param）。后端接收到请求后，将请求的参数封装为 QueryPageParam 对象，再通过 PageHelper 开启分页，并设置当前的页码和页面大小，然后利用 recipeOpService 对象的 getUserStarList 方法获取用户的点赞列表。如果点赞列表不为空，则根据这些点赞列表 id 调用 recipeService 的方法逐个获取对应的菜谱列表，并使用 PageInfo 将这些点赞菜谱列表进行封装，

最终返回给前端。前端在接收到请求的成功响应后，更新页面的总数据量、当前页面数据量并计算展示食谱列表的行数和列数，再将食谱列表数据赋值给 `recipes`。`recipes` 会通过 `v-for` 指令和上述模块中的 `OneShowRecipe` 组件进行渲染食谱列表，当用户点击某个食谱项时通过 `$router.resolve` 跳转新的页面展示食谱详情数据。

### 6.3.3.2 用户主页设计与实现

用户可以在食谱详情页面、社区交流页面通过点击跳转到用户主页。此点击事件会触发 `toUserCenter` 函数，该函数使用 `$router.resolve` 方法构造一个路由信息对象，指定了要前往的路径为“/usercenter”并携带了一个参数为 `userId`，再通过 `window.open` 方法以新窗口的形式打开这个构建好的路由链接。该路由对应了 `UserCenter` 组件。`UserCenter` 组件通过 `$route.query.userId` 来获取传递的用户 id，可以在用户主页查看该用户的一些基本信息、点赞、收藏食谱等。用户主页和个人主页的子组件是共用的，只需传递 `userId` 便可以获取相关用户的数据并将其渲染到页面上，此处不再赘述。

## 6.4 后台管理系统详细设计与实现

后台管理系统主要完成对用户端系统数据进行更新和维护，主要包含用户管理模块、权限管理模块、食谱管理模块和社区交流模块。本节将对这四大模块进行详细设计，并描述其中涉及的类和接口。

### 6.4.1 用户管理模块

用户管理模块负责对注册到系统中的用户信息进行维护和更新，管理员在该模块中能够编辑用户的个人基本信息，为用户重置密码和删除不活跃的用户，也可以新增用户，并为该用户设置初始密码。在新增用户时，管理员可以输入用户的用户名、性别、年龄、电话、邮箱和住址，也可选择用户的角色进行添加。

在用户管理模块中，管理员可以通过输入框、日期选择器，通过用户的姓名、用户名等内容筛选符合条件的用户信息。当没有该用户的具体信息时，管理员可以通过新增用户界面进行添加用户的操作，并为该新增用户分配所属角色。当进行添加时，系统会根据输入的用户名判断该用户名是否存在，如果存在则返回必要的提示信息；如果不存在可以执行新增用户信息的操作。筛选用户时，如果用户存在，可以为用户进行重置密码或删除用户。上述流程执行完毕时，系统会自动调用函数重新加载和渲染页面数据。

如图 6-12 所示，为用户管理模块主要的流程图。

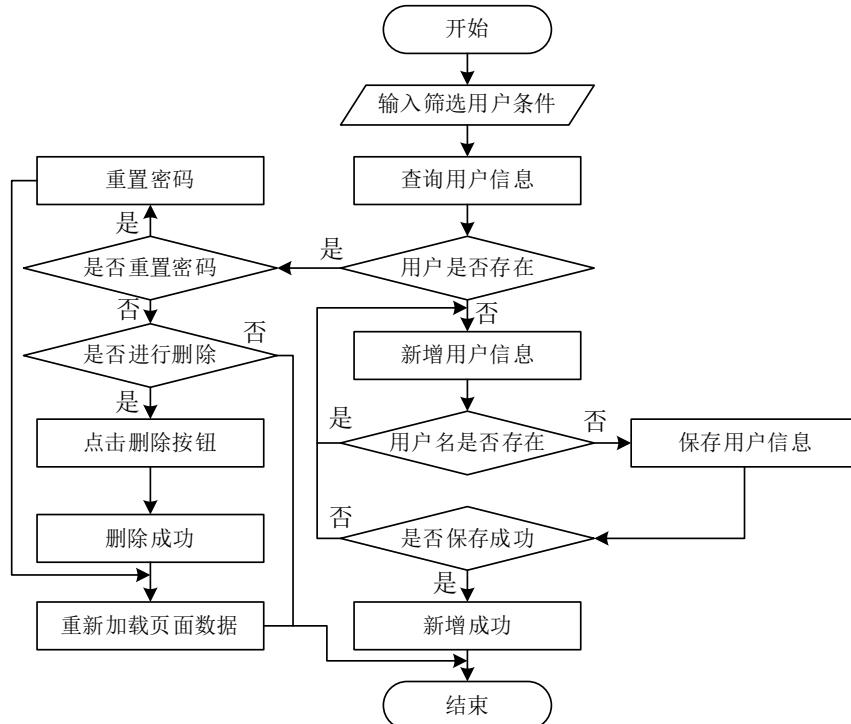


图 6-12 用户模块流程图

用户管理模块所涉及的类有控制器 UserController、用户接口类 UserService、用户接口实现类 UserServiceImpl、用户实体类 User。如表 6-6 所示，为用户管理模块使用的接口表。

表 6-6 用户管理模块接口表

URL 地址	参数类型	描述
POST /user/getUsersByCondition	QueryPageParam q	分页模糊查询用户
POST /user/getAllUsers	QueryPageParam q	分页查询所有用户
PUT /user/update	User user	更新用户信息
DELETE /user/delete	Integer userId	删除 id 为某值的用户
POST /user/put	User user	添加用户信息
POST /user/resetpass	Map<String, String> ms	重置用户登录密码
POST /user/getOneById	Integer userId	获取某个用户信息

在表 6-6 中，包含了查询、更新、删除、添加用户信息。其中，在分页模糊查询中，传入的参数类型为 QueryPageParam，该对象中有两个变量分别表示分页大小和页号。另外一个变量的类型为 HashMap<String, Object>。该类型的参数用于在

controller 层向 service 层传递使用，通过这样的参数传递，在 Mapper 的 XML 文件中配置 paramType 属性为 java.util.Map，即可在 SQL 语句中使用#{key}访问 HashMap 变量中 key 所对应的 value 值。因此，在 XML 文件中可以利用该变量参数使用 trim、if、where 等标签来动态生成 SQL 语句，以完成应对不同的参数构造的情况。

#### 6.4.2 权限管理模块

权限管理模块可以对后台管理系统的角色进行添加、编辑和删除操作，对后台系统的菜单进行新增、更新和删除，同时也能够为用户分配不同的角色，为角色分配其可见的菜单。在分配角色菜单信息时，系统采用树形结构的展现方式，管理员可以通过点击选择框进行分配。在新增和编辑菜单时，管理员可以对菜单的昵称、URL 地址、icon 图标等信息进行修改。本模块的主要作用是对后台管理系统的管理员所分权限有所限制，使不同的角色呈现不同的菜单界面，只能使用显示的菜单功能，增强了后台系统的安全性和稳定性，保证了用户端系统的可靠性和稳定性。

如图 6-13 所示，为权限管理模块流程图。拥有本模块管理权限的管理员可以通过输入筛选匹配值来查找相应的角色，并且可以查看某个角色被分配的菜单详情。如果某角色没有管理员想要分配的菜单时，通过菜单界面查找是否存在该菜单。如果不存在该菜单，则通过点击分配菜单按钮进行分配，并设置菜单的名称、URL 地址、icon 等相关信息，然后再为该角色分配相应的菜单。

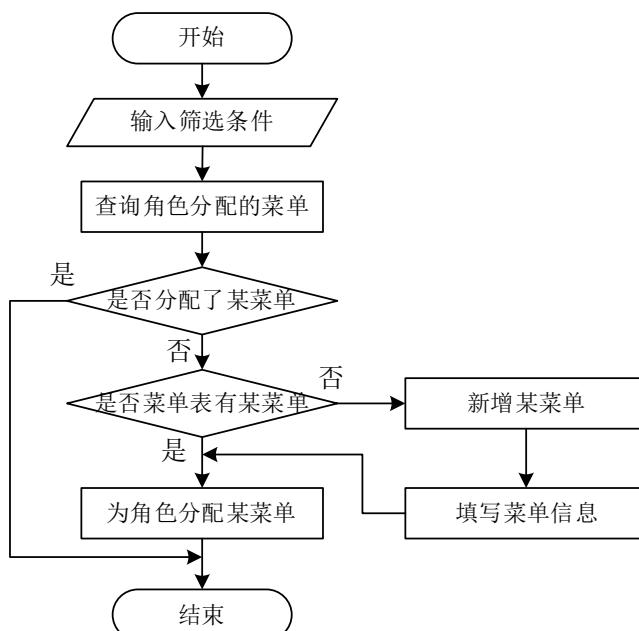


图 6-13 权限管理模块流程图

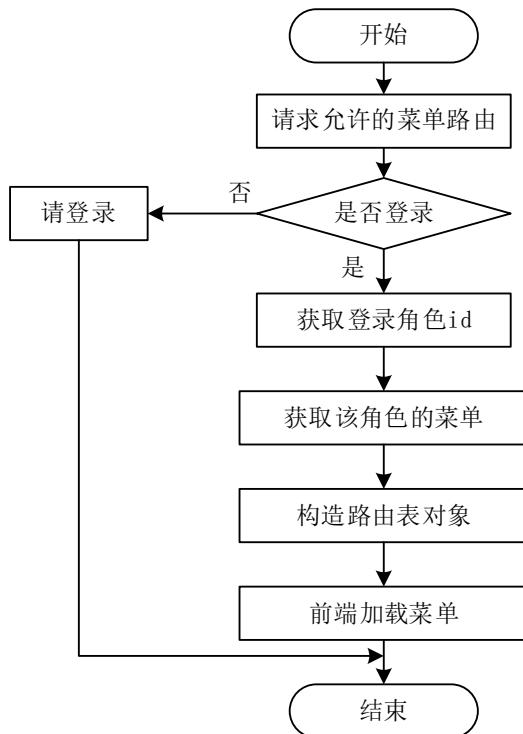


图 6-14 动态加载路由流程图

权限管理模块还承担了管理员登录后台系统后动态加载路由的功能。该功能为登录到本系统的管理员动态渲染路由表和菜单，让管理员角色不可见的菜单项隐藏。如图 6-14 所示，为前端动态加载路由的流程图。当管理员登录到系统后，前端会向后端发送请求获取允许的菜单路由，后端会检查请求所携带的令牌（JWT）是否有效，如果无效则说明管理员登录权限已过期需要重新登录；如果有效，则后端会通过携带的令牌获取管理员的个人信息从而确定当前管理员的角色 id。其次，后端通过角色 id 查询该角色可见的菜单路由，最终将其构造为前端路由表要求的对象格式并返回给前端。前端 router 会根据这个路由对象动态加载到页面中，进而实现动态路由的效果。其中，返回的菜单路由对象的构造方式为 Role 对象中含有 children 变量，该变量的类型为 List<Role>，因此前端在渲染子菜单时可以访问父节点的 Children 成员变量即可。

本模块所涉及的类有控制器 RoleController、菜单控制器 MenuController、角色菜单控制器 RoleMenuController、角色接口类 RoleService、菜单接口类 MenuService、角色接口实现类 RoleServiceImpl 等类。

如表 6-7 所示，为权限管理模块接口表，其中包含了获取角色信息、模糊查询角色信息等接口。

表 6-7 权限管理模块接口表

URL 地址	参数类型	描述
GET /role/	—	获取所有角色信息
POST /role/getRolesByCondition	QueryPageParam p	模糊分页查询角色信息
PUT /role/update	Role role	修改一个角色信息
DELETE /role/delete	Integer id	删除 id 为某值的角色
POST /role/add	Role role	添加一个角色信息
GET /menu/getFirstLevel	—	查询所有父节点菜单
PUT /menu/update	Menu menu	更新菜单信息
POST /menu/add	Menu menu	添加菜单信息
DELETE /menu/delete	Integer menuId	删除 id 为某值的菜单
GET /roleMenu/get	Integer roleId	获取角色可见的菜单列表
POST /roleMenu/update	HashMap<String, Object> h	更新角色可见的菜单列表

当管理员进入角色页面时，会渲染 Role.vue 组件，该组件渲染执行 mounted 钩子函数时，会触发 initData 函数。该函数首先将 loading 状态置为 true，显示加载状态。其次，通过 postRequest 方法发送 POST 请求，请求参数为封装的 page 对象，请求的 URL 为 “/api/role/getRolesByCondition” 。后端在接收到发送的请求后，通过 roleService 对象调用 Mapper 层动态生成的 SQL 语句得到类型为 List<Role> 的角色列表，再通过 PageInfo 对象进行封装后返回给前端。前端在接收到请求的成功响应后，将相关的角色内容赋值给 roles 对象，最终通过 el-table 渲染在页面上。

管理员点击为某角色设置菜单列表时，会触发 toEditMenuForRole 函数，传递的参数为角色 id。该函数将 editMenuForRoleDialog 置为 true，随后显示修改角色菜单列表的对话框，其次通过 \$nextTick 设置内置函数，当页面的渲染工作完成后再执行。该内置函数是通过 \$refs 获取子组件 EditRoleMenu 的 ref，再通过此 ref 调用 setRoleId 函数，将角色的 id 赋值给该子组件。子组件接收到角色 id 后，会调用 getAllMenus 函数将所有的菜单渲染到 el-tree 树形组件中，再通过 getRequest 方法发送 GET 请求，请求的 URL 为 “/api/roleMenu/get” ，请求的参数为角色 id。后端接收到发送的请求后，获取角色 id，并创建 QueryWrapper 对象设置“id”字段为角色 id，再使用 roleService 查询与该角色相关的菜单列表，最终返回给前端。前端接收到成功的响应后，将后端

返回的数据赋值给 menus，并通过循环将该角色可见的菜单 ID 存储到 checkedMenu 数组中，el-tree 子组件会根据该数组信息渲染哪些选择框被选中。

管理员可以在上述对话框中勾选某些选择框来编辑该角色可见的菜单。管理员点击保存按钮会触发 updateRoleMenu 函数，该函数会向“/api/roleMenu/update”发送 POST 方法，参数为角色 id 和可见菜单列表。后端使用 HashMap<String, Object>接收参数，并提取角色 id 和可见菜单列表，其次创建一个 QueryWrapper 对象，设置“id”为角色 id，再利用 roleMenuService 将该角色所可见的菜单全部删除，并将前端传输的可见菜单列表循环构建 RoleMenu 对象，再将其保存到数据库中。前端接收到保存成功的响应后，会使用 location.reload 函数刷新当前页面更新权限。

#### 6.4.3 食谱管理模块

管理员可以通过食谱管理模块对食谱信息进行检索，并且能够对食谱进行编辑和删除操作。此外，管理员可以通过 el-table 的表头对点赞、收藏和浏览记录进行升序和降序的筛选和查询。本模块所涉及的主要类有控制器 RecipeController、食谱接口类 RecipeService、食谱接口实现类 RecipeServiceImpl、食谱实体类 Recipe、食谱操作接口类 RecipeOpService、食谱操作接口实现类 RecipeOpServiceImpl、RedisUtils。

表 6-8 食谱管理模块接口表

URL 地址	参数类型	描述
GET /recipe/getAllCategories	—	获取所有类别
POST /recipe/getByCondition	QueryPageParam p	模糊分页查询
DELETE /recipe/delete	Integer id	删除 id 为某值的食谱
POST /recipe/add	Recipe recipe, Principal p	添加食谱信息
POST /recipe/update	Recipe recipe	更新食谱信息
POST /recipe/getAll	QueryPageParam p	分页查询食谱信息
POST /recipe/getOne	Integer id	查询 id 为某值的食谱

如表 6-8 所示，为食谱管理模块主要接口表，其中包含了获取所有类别，模糊查询食谱，删除 id 为某值的食谱等。特别的，在模糊查询分页中，利用了数据类型 HashMap<String, Object> 进行传值，其中，starNumOrder 存储了 star\_num 字段排序方式为“DESC”还是“ASC”，在 Mapper 中只需使用 \${starNumOrder} 便可以通过 trim、if 标签动态生成某个字段以升序排序还是降序排序。

#### 6.4.4 社区交流模块

管理员在社区交流模块中可以进行筛选用户发布的主题，通过点击跳转主贴详情页，编辑主贴的内容和图片，删除主贴等操作。该模块涉及的主要类有主贴控制器 TopicController、主贴接口类 TopicService、主贴回复接口类 TopicReplyService、主贴接口实现类 TopicServiceImpl、主贴回复接口实现类 TopicReplyServiceImpl 以及所包含的实体类。如表 6-9 所示，为社区交流模块主要涉及的接口表。

表 6-9 社区交流模块接口表

URL 地址	参数类型	描述
POST /topic/getByCondition	QueryPageParam p	模糊分页查询主贴
GET /topic/getOneTopic	Integer topicId	查询 id 为某值的主贴
POST /topic/update	Topic topic	更新主贴内容
POST /topic/getByUser	QueryPageParam q	查询某用户发布的主贴
DELETE /topic/delete	Integer topicId	删除 id 为某值的主贴

#### 6.5 饮食推荐模块

本模块主要负责收集用户的喜好类别，并通过用户的历史点赞、收藏记录和用户选择的喜好类别进行推荐。本模块可以为新用户和老用户进行推荐。系统通过新用户选择的喜好类别进行推荐，而对于老用户，系统通过分析老用户的历史交互记录，调用模型进行推荐。

##### 6.5.1 为新用户进行推荐

用户进入食谱推荐页面后，系统会提示用户请选择喜好的类别，其中包含工艺、口味、所需时间等信息。当用户选择完毕后，系统会将用户的喜好类别存储在数据库中，并利用 Neo4j 数据库给出这些喜好类别基于入度进行推荐和基于相似度进行推荐。用户也未必一定会喜欢推荐的结果，系统希望能够尽可能添加随机因素来充分挖掘用户的兴趣。因此，在获取基于入度和基于相似度的推荐时，系统会将两者的结果依照推荐顺序随机抽取两者的推荐，以最大限度为用户展示推荐的解决。当新用户进入食谱推荐页面时，页面会优先显示用户选取喜好类别的界面，当用户选择喜好类别后，页面会自动收起选择界面，并展示为该用户推荐的食谱结果。

### 6.5.2 为老用户进行推荐

用户进入食谱推荐页面时，系统会分析得到用户的点赞和收藏食谱，并将这些食谱信息输入到 RippleNet 模型中，最终得到模型推荐的结果，其次再将基于入度和基于相似度的推荐结果相加，得到向用户展示的推荐结果。由于获取推荐结果的程序可能需要耗费一定的时间，所以当获取到某用户的推荐结果时，将结果放置在 Redis 中，并设置为 1 天的有效期，以便后续能够快速找到用户的推荐列表。

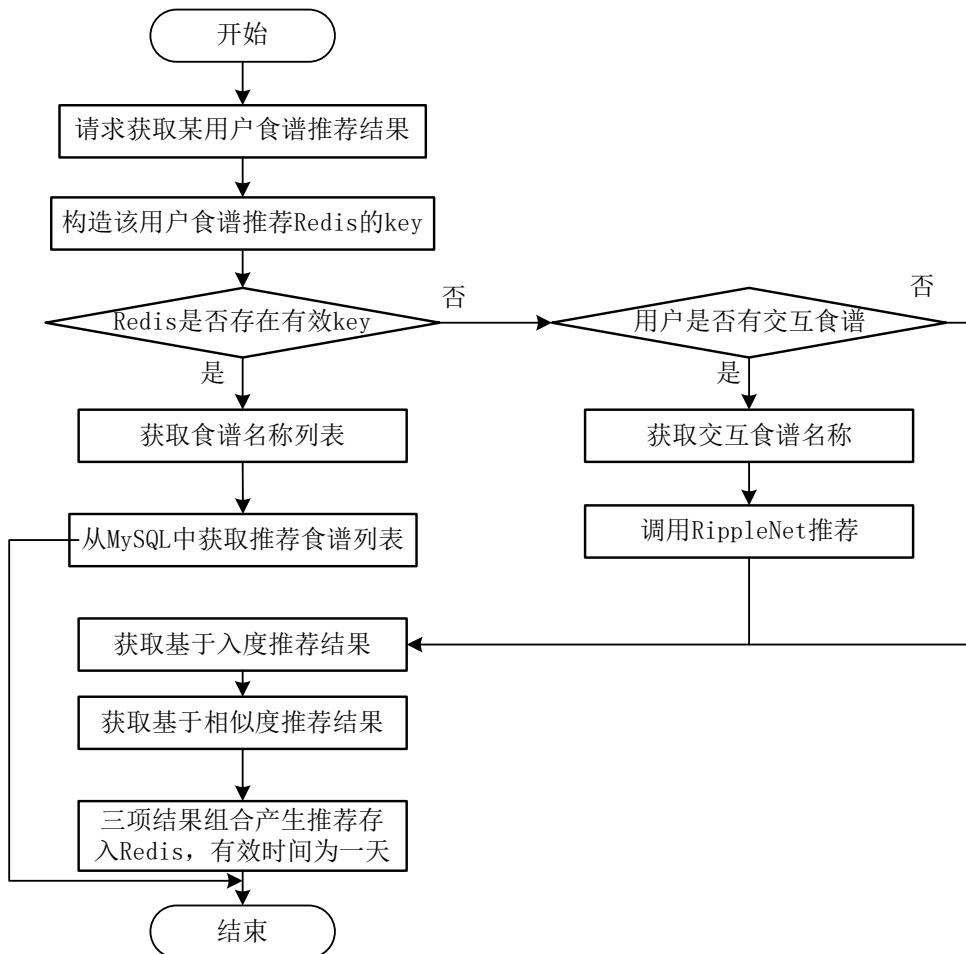


图 6-15 食谱推荐流程图

如图 6-15 所示，为食谱推荐的流程图。用户请求获取饮食推荐时，后端会构造该用户在 Redis 中存放食谱推荐结果的 key，如果存在且有效，则取出食谱名称列表，并依次循环名称列表得到食谱列表，返回给前端；如果不存在则需要判断用户是否有交互记录，如果有则取出交互记录的食谱名称，并调用 RippleNet 模型获取模型推荐结果，再通过获取用户的喜好类别，基于入度进行推荐结果和基于文本相似度进行推荐结果，并将结果存入数据库中，设置有效时间为 1 天。

对于操作 Neo4j 数据和调用 RippleNet 模型，系统使用了 Python 和 Flask 框架搭建相关的 Web 服务，并设置 ip 地址和端口号。Java 通过 RestTemplate 类发送 POST 或 GET 请求来使用 Flask 暴露的接口。如表 6-10 所示，为本模块使用的相关接口表。

表 6-10 饮食推荐模块接口表

URL 地址	参数类型	描述
GET /recommend/getNewUser	Principal principal	为新用户推荐
GET /recommend/getOldUser	Principal principal	为老用户推荐
GET /recommend/getRecommend	Principal principal	用户获取推荐
POST /getByCates (Flask Web 端)	String str	通过喜好类别推荐
POST /getBySimilar	String str	通过相似度推荐
POST /getByRippleNet	String str	通过 RippleNet 模型推荐

## 6.6 成功展示

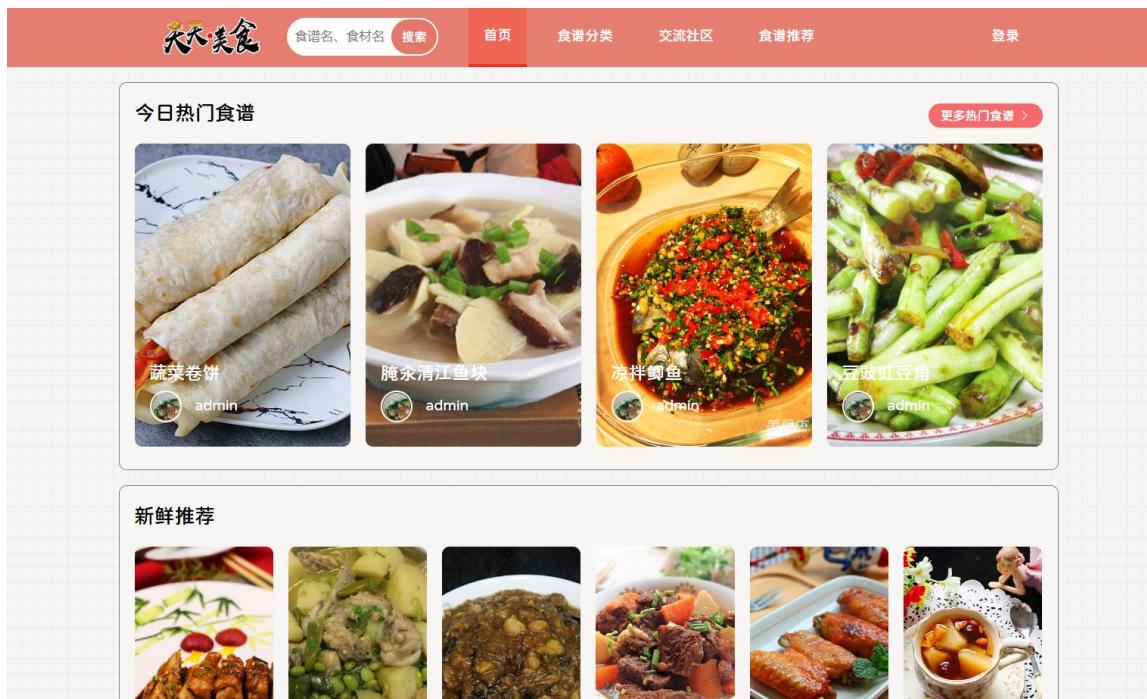


图 6-16 用户端食谱主页

如图 6-16 所示，为用户端食谱主页。用户可以在该页面查看今日热门食谱和食谱列表，可以通过点击某个食谱小卡片进入该食谱的详情页面。在食谱主页，用户可以查看今日热门食谱，并可以在右侧点击更多热门食谱按钮进入食谱筛选页面，其次用户往下滑动页面可以看到其它食谱信息。

## 燕山大学本科毕业设计（论文）

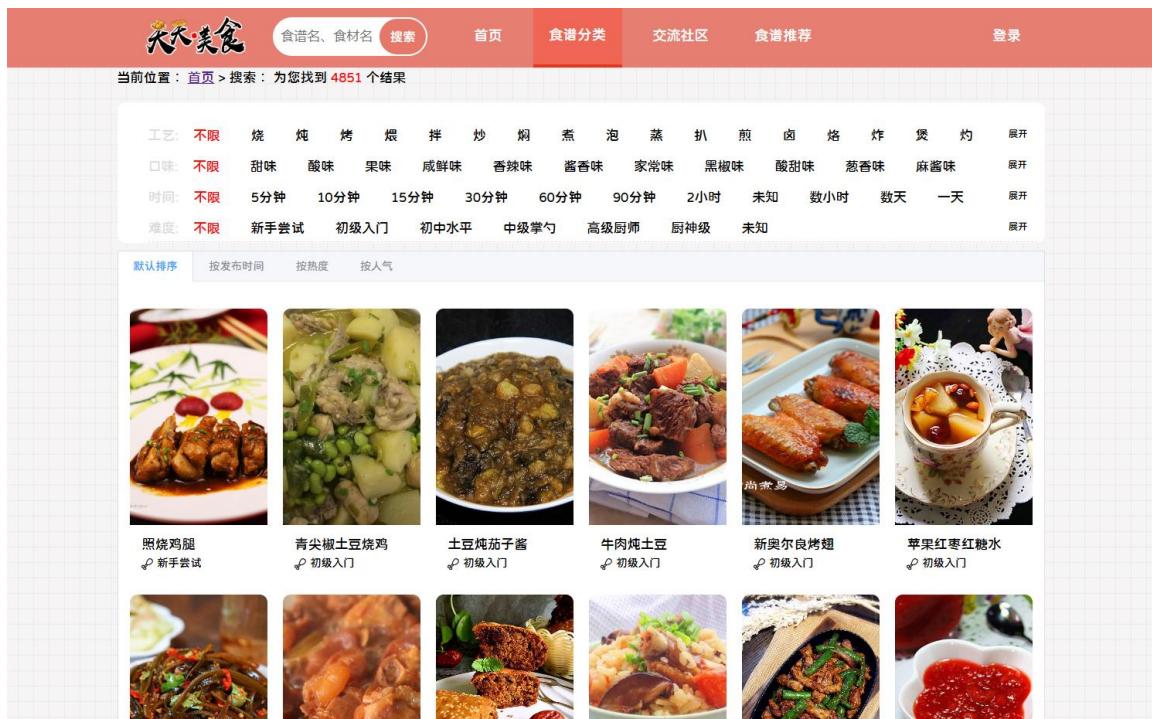


图 6-17 食谱筛选页

如图 6-17 所示，为食谱的筛选页。用户可以通过不同的类别和关键词进行筛选食谱，筛选类别有工艺、口味、时间和难度。此外，在筛选出的食谱中，用户可以根据默认排序、发布时间、热度和人气对食谱列表进行排序，从而获取想要的食谱信息。

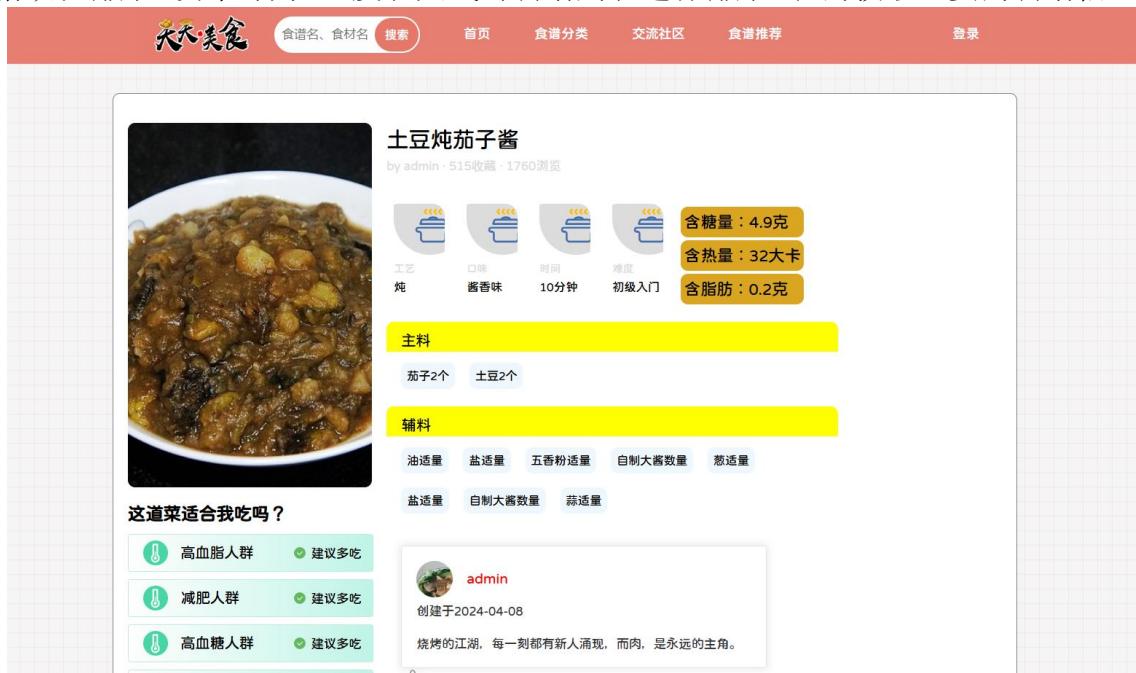


图 6-18 食谱详情页

如图 6-18 所示，为食谱详情页。用户可以在该页面查看食谱的详细信息。

## 第 6 章 系统的详细设计与实现

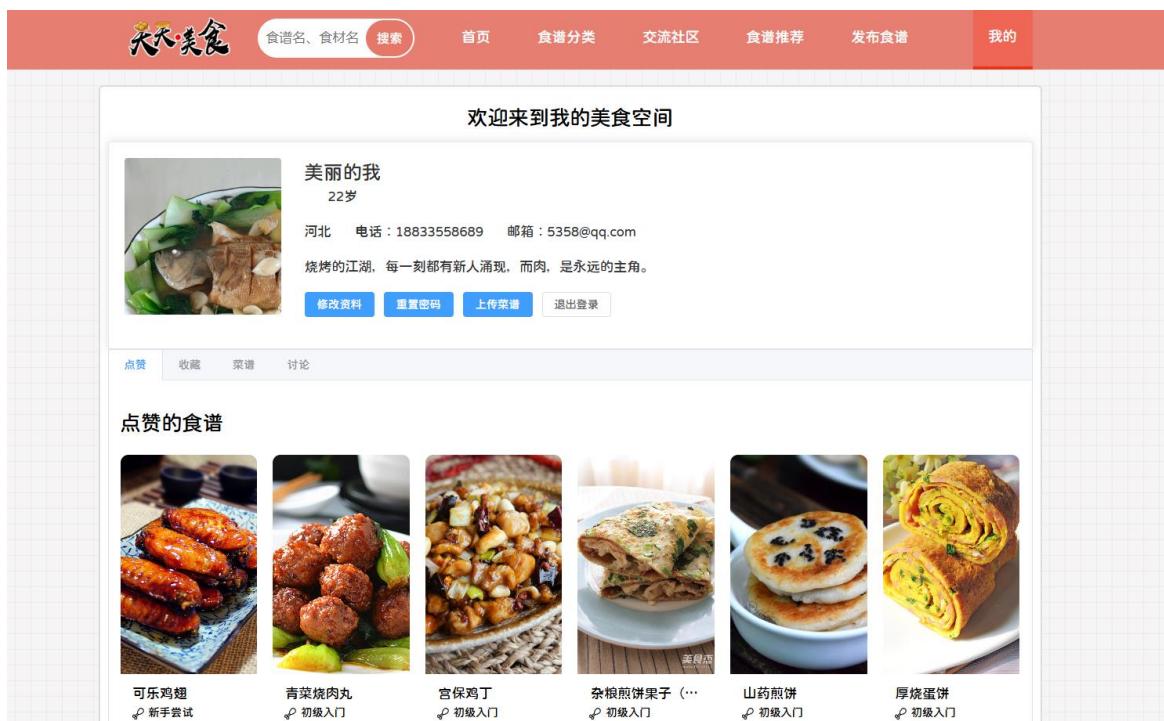


图 6-19 个人中心页

如图 6-19 所示, 为用户的个人中心页面。用户可以在该页面中进行修改资料、重置密码、上传菜谱和退出登录的操作。同时, 用户可以在该页面查看自己所点赞、收藏和发布的食谱, 也能够查看自己所发布的主贴内容。

图 6-20 发布食谱页

如图 6-20 所示, 为发布食谱页。登录用户可以在本页发布自己制作的食谱。

## 燕山大学本科毕业设计（论文）

如图 6-21 所示，为用户端的食谱推荐页。新用户在进入该主页时，可以通过选择自己所喜爱的工艺和口味获得相应的推荐。当用户在网站中活跃一段时间后，系统可以根据用户的历史交互记录为用户推荐潜在的兴趣食谱。

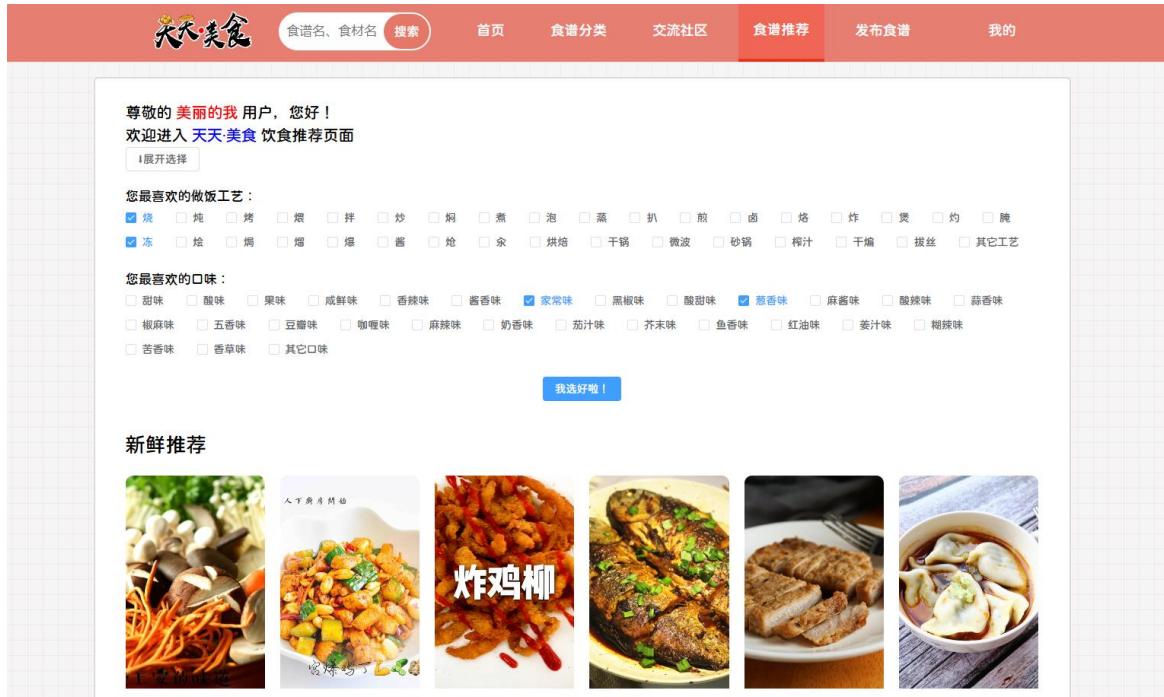


图 6-21 食谱推荐页

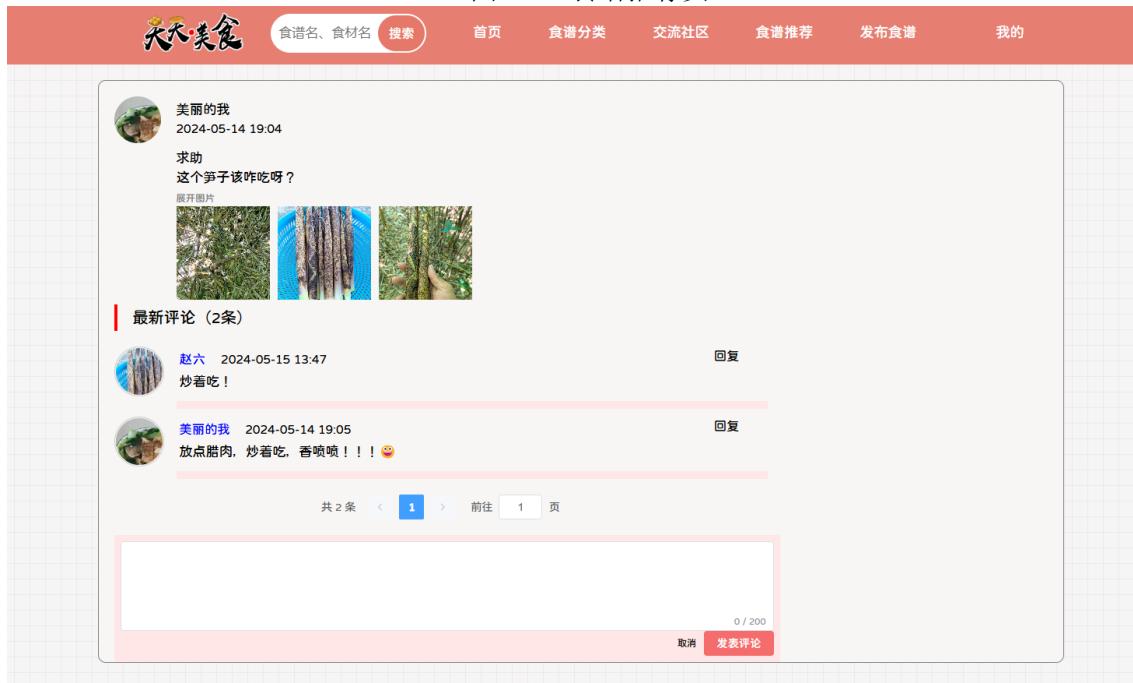


图 6-22 主贴详情页

如图 6-22 所示，为主贴详情页。用户可以在本页面查看其它用户发布的主贴信息和回复信息，并能够通过点击用户的头像进入该用户的主页。

## 第 6 章 系统的详细设计与实现

The screenshot shows a table of recipes. Here is a sample of the data:

名称	主食材	辅食材	工艺	口味	所需时间	难度	点赞	浏览	收藏	操作
照烧鸡腿	鸡腿3根	酱油1勺 白砂糖...	烧	咸鲜味	15分钟	新手尝试	8	3899	1662	[编辑] [删除]
青尖椒土豆烧鸡	鸡翅根500克 土豆...	八角少许 回香...	烧	香辣味	60分钟	初级入门	0	2980	404	[编辑] [删除]
土豆炖茄子酱	茄子2个 土豆2个	油适量 盐适量 ...	炖	酱香味	10分钟	初级入门	0	1738	513	[编辑] [删除]
牛肉炖土豆	牛腩500克 土豆...	洋葱适量 大葱1...	炖	咸鲜味	90分钟	初级入门	0	6831	455	[编辑] [删除]
新奥尔良烤翅	鸡翅中500克	盐2克 蜂蜜2汤...	烤	香辣味	15分钟	初级入门	0	6676	97	[编辑] [删除]
苹果红枣红糖水	苹果一个 红枣...	桂圆半个 红糖...	煮	甜味	2小时	初级入门	0	4978	5	[编辑] [删除]
油拌香辣海带丝	海带丝一份	小米辣六七个【...	拌	香辣味	30分钟	初中水平	0	626	5	[编辑] [删除]
红剁猪蹄 (我的...)	猪蹄	葱姜白芷2片 ...	炒	家常味	5分钟	新手尝试	0	1888	238	[编辑] [删除]

图 6-23 食谱管理页

如图 6-23 所示，为后台系统的食谱管理页。管理员可以通过该页面对食谱进行编辑和删除操作。

The screenshot shows a table of posts. Here is a sample of the data:

序号	标题	内容	图片	发布用户	创建时间	更新时间	类型	操作
1	真不错	自己在家做秦烧鲈鱼，...		美丽的我	2024-05-14 17:57	2024-05-14 17:57	美食随手拍	[编辑] [删除]
2	求助	这个笋子该咋吃呢？		美丽的我	2024-05-14 19:04	2024-05-14 19:04	厨艺交流	[编辑] [删除]
3	123	测试		赵六	2024-05-15 13:48	2024-05-15 13:48	美食随手拍	[编辑] [删除]

图 6-24 社区交流管理页

如图 6-24 所示，为后台系统的社区交流管理页面。管理员可以通过该页面对用户发布的主贴内容进行编辑和删除操作。

## 6.7 本章小结

本章分模块对本项目进行了详细设计，并描述了系统主要模块的实现方法和步骤，给出了各个功能模块所使用的接口、参数类型和描述，确保系统的每个部分能够正常完成基本功能，保证系统能够稳定安全地运行。



## 第 7 章 系统测试

系统测试是软件开发生命周期中至关重要的环节之一，是系统能够正常稳定运行的保障。在系统正式应用前，对各功能模块进行测试是为了能够发现程序的缺陷和漏洞，及时对系统进行修正和改进，以保证系统的可用性和完整性<sup>[20]</sup>。

### 7.1 系统测试环境说明

如表 7-1 所示，为本系统的测试环境说明表。

表 7-1 测试环境表

配置项	环境或版本
处理器	Core i7
内存	16GB
操作系统	Windows 11
MySQL	8.0.26
JDK	1.8
SpringBoot	3.4.1
浏览器	Edeg 浏览器、Google Chrome

### 7.2 系统功能测试用户

本节将在需求分析和功能模块设计的基础上，分模块对本系统进行测试，保证各个模块能够符合预期结果，确保各个功能模块能够完成响应的功能。

#### 7.2.1 用户端功能测试

(1) 食谱浏览功能测试用例如表 7-2 所示。

表 7-2 食谱浏览功能测试用例

测试项目	预期输出	测试结果
未登录用户进入主页	在主页显示食谱列表	符合预期

表 7-2（续表）

测试项目	预期输出	测试结果
未登录用户切换页面	食谱列表进行换页，换页显示加载动画	符合预期
未登录用户切换到尾页	食谱列表切换，最后一行不占满	符合预期
未登录用户通过“土豆”关键词进行搜索食谱	跳转搜索页，显示有关土豆食谱列表内容	符合预期
未登录用户通过多个类别进行筛选	食谱类别筛选正确，显示食谱	符合预期
未登录用户点击进入食谱详情页	跳转到食谱详情页，查看食谱的相关步骤、工艺等	符合预期
未登录用户进行点赞、收藏食谱	提示未登录，请登录，无法进行点赞操作	符合预期
通过登录界面进行登录，登录用户名为 ceshi1	成功登录，顶部显示“我的”和“发布食谱”	符合预期
用户 ceshi1 进行点赞（收藏）食谱操作	用户在食谱详情页面看到成功点赞（收藏）状态	符合预期
刷新页面	食谱点赞显示点赞（收藏）状态	符合预期

(2) 社区交流功能测试用例如表 7-3 所示。

表 7-3 社区交流功能测试用例

测试项目	预期输出	测试结果
未登录用户查看主贴列表	显示主贴列表内容和图片	符合预期
未登录用户搜索主贴内容	显示搜索内容相关的主贴	符合预期
未登录用户选择主贴类型	显示选择的类型相关的主贴	符合预期

表 7-3 (续表)

测试项目	预期输出	测试结果
未登录用户切换页号	主贴列表切换页面	符合预期
未登录用户发布主贴	发布框显示“请登录”	符合预期
用户 ceshi1 发布主贴内容和图片	当前页面出现 ceshi1 发布的主贴	符合预期
用户 ceshi2 进入 ceshi2 发布的主贴	用户 ceshi2 看到主贴内容和图片	符合预期
用户 ceshi2 回复主贴内容	当前用户看到 ceshi2 的回复内容	符合预期

(3) 个人中心测试用例如表 7-4 所示。

表 7-4 个人中心功能测试用例

测试项目	预期输出	测试结果
未登录用户登录账号 ceshi1	成功登录账号 ceshi1	符合预期
点击“我的”进入个人主页	显示 ceshi1 的个人信息和头像	符合预期
用户 ceshi1 点击头像进行修改	用户 ceshi1 头像进行了修改	符合预期
用户 ceshi1 修改个人资料	ceshi1 的个人资料修改成功	符合预期
用户 ceshi1 修改密码	ceshi1 退出登录后用新密码登录成功	符合预期
用户 ceshi1 点赞某个食谱后，查看个人的点赞（收藏）记录	查看到有该点赞（收藏）记录	符合预期
用户 ceshi1 发布个人食谱	在食谱列表中查看到用户 ceshi1 发布的食谱	符合预期
用户 ceshi1 在个人中心删除发布的某食谱	用户 ceshi1 发布食谱列表移除了删除的食谱	符合预期
用户 ceshi1 发布某主贴	用户 ceshi1 获取到发布主贴内容	符合预期

表 7-4 (续表)

测试项目	预期输出	测试结果
用户 ceshi1 通过点击进入用户 ceshi2 的个人主页	成功进入用户 ceshi2 的个人主页	符合预期

## 7.2.2 管理员端功能测试

(1) 用户管理功能测试用例如表 7-5 所示。

表 7-5 用户管理功能测试用例

测试项目	预期输出	测试结果
管理员 admin1 修改用户 ceshi1 的个人信息	用户 ceshi1 的个人信息被修改	符合预期
管理员 admin1 删除用户 ceshi1	用户列表移除了用户 ceshi1	符合预期
管理员 admin1 点击按钮新增用户 ceshi2	用户列表添加了用户 ceshi2	符合预期
管理员 admin1 重置 ceshi2 的登录密码	ceshi2 使用新密码登录成功	符合预期

(2) 权限管理功能测试如表 7-6 所示。

表 7-6 权限管理功能测试用例

测试项目	预期输出	测试结果
管理员 admin1 新增 role1 和 role2	角色列表添加了 role1 和 role2	符合预期
管理员 admin1 删除角色 role2	角色列表移除了角色 role2	符合预期
管理员 admin1 为管理员 admin2 分配角色 role1	管理员 admin2 的角色为 role1	符合预期
管理员 admin1 为角色 role1 分配菜单“用户管理”和“食谱管理”	管理员 admin2 登录后台系统，显示“用户管理”和“食谱管理”	符合预期

表 7-6 (续表)

测试项目	预期输出	测试结果
管理员 admin1 取消 role1 的“食谱管理”权限	管理员 admin2 刷新界面，不显示“食谱管理”菜单	符合预期

(3) 食谱管理功能测试如表 7-7 所示。

表 7-7 食谱管理功能测试用例

测试项目	预期输出	测试结果
管理员 admin1 查看食谱列表	食谱列表可进行分页切换	符合预期
管理员 admin1 编辑某食谱	管理员进入编辑页面，保存成功	符合预期
管理员 admin1 删除某食谱	某食谱从食谱列表中移除	符合预期
管理员 admin1 通过“土豆”、其它类型选项进行筛选	食谱列表显示筛选结果	符合预期

(4) 社区交流模块功能测试如表 7-8 所示。

表 7-8 社区交流模块功能测试用例

测试项目	预期输出	测试结果
管理员 admin1 编辑主贴信息	主贴列表信息变化	符合预期
管理员 admin1 删除主贴信息	主贴列表中移除了删除的主贴	符合预期
管理员 admin1 发布主贴内容	主贴列表中新增了主贴内容	符合预期
管理员 admin1 编辑主贴信息	主贴列表中主贴内容发生变化	符合预期

### 7.3 本章小结

本章在上文需求分析和详细设计的基础上，描述了测试系统的环境，并从用户端和管理员端两方面分模块编写测试用例，并进行了详细测试。通过详细测试，确保了系统能够稳定正常运行。



## 结 论

随着人们日益增长的美好生活需要和不平衡、不充分的饮食分享与交流领域发展之间的矛盾日益显著，满足用户需要的饮食推荐与交流分享系统能够充分让用户投入其中，增强用户间的交流，刺激食品消费，为饮食推荐行业发展提供动力。

本项目通过深入分析美食杰网站的构成，构建本系统的食谱数据集和用户-食谱偏好数据集，使用 RippleNet 模型挖掘用户对食谱的潜在喜好，为用户推荐食谱。其次，本系统实现了良好的食谱分享平台，支持用户查看发布在本系统的食谱，为用户提供了良好的筛选和查看食谱界面。用户也可以在本平台中发布食谱和分享美食心得，增强用户间的交流，打造良好的网站交流氛围。本系统利用 Neo4j 内置图算法进行食谱推荐，推荐结果可解释性强，推荐过程灵活高效。

随着互联网技术的不断发展和进步，食谱推荐领域有很多令人惊奇的研究。通过多模态的食谱推荐是当前的研究热点<sup>[2]</sup>。基于 GPT 的食谱推荐系统更是目前最有吸引力的研究方向，能够通过聊天的方式快速提取用户的日常喜好和身体健康状况，从而实现更高效、更便捷的个性化推荐。当然，本项目还存在许多不足之处。

(1) 在食谱获取的过程中，未考虑到食材和食谱的营养元素，虽然实现了一定的饮食推荐，但推荐结果未必符合营养学层面的推荐，下一步可以将食材的营养元素纳入知识图谱中，实现更全面的推荐。

(2) 食谱数据集质量和用户-食谱偏好数据集质量不佳，导致推荐模型的准确率较差。后续可以通过不断完善自身数据集的构成，提高数据集的质量，优化模型参数，不断提高模型的准确率。

(3) 由于毕业设计时间有限，饮食知识图谱的构建未严格按照知识图谱的标准进行设计和构建，仅是基于提前设定的图结构和关系提取实体和关系，构建相关的数据集，模型扩展能力有限，不具备知识推理能力。

(4) 本项目数据集结构单一，关系数量较少，使用 RippleNet 模型的准确率不高，学习难度较大，性能欠佳。

接下来的项目的实施和研究会依照 ml-1m 的规范，着重构建更加完整的知识图谱，提高食谱知识图谱的质量，发现更多食谱间和属性间的联系，从而提高准确率。



## 参考文献

- [1] 王佳仪. 营养健康食品呈现消费新趋势[N]. 中国食品报,2024-03-13(006).
- [2] 江山. 基于知识图谱的个性化食谱推荐算法研究与实现[D].浙江理工大学,2024.
- [3] Marin J, Biswas A, O li F, et al. Recipe1m+: A dataset or learning cross-modal embeddings for cooking recipes and ood images[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(1): 187-203.
- [4] Bushra N, Hasan M. uicklyCook: A User-Friendly Recipe Recommender[C]//Machine Learning and Metaheuristics Algorithms, and Applications: First Symposium, SoMMA 2019,Trivandrum, India, December 18–21, 2019, Revised Selected Papers 1. Springer Singapore,2020: 245-254.
- [5] Haussmann S, Seneviratne O, Chen Y, et al. FoodKG: a semantics-driven knowledge graph for ood recommendation[C]//The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18.Springer International Publishing, 2019: 146-162.
- [6] Li M, Li L, Xie , et al. MealRec: A Meal Recommendation Dataset[J]. arXiv preprint arXiv:2205.12133, 2022.
- [7] 仲亮靓. 基于饮食知识图谱的食谱推荐系统的设计与实现[D]. 东南大学, 2019.
- [8] 董洪伟. 基于知识图谱的菜品推荐系统[D]. 北京: 北京林业大学, 2020.
- [9] 王崟.浅谈 Java web 框架课程中 Maven 的教学内容和设计[J].电脑知识与技术,2021,17(24):85.
- [10] 于双龙. 基于 SpringBoot 的 APM 系统的设计与实现[D].北京交通大学,2023.
- [11] 刘慧娟. 基于 SpringBoot 的民主测评系统的设计与实现[D].北京邮电大学,2024.
- [12] 展维维. 基于 Redis 的分布式存储系统设计与应用[D].南京大学,2021.
- [13] 曾国林,傅秀芬.一种新的数据库连接池模型的研究[J].计算机与数字工程,2011,39(02):163-166.
- [14] 欧阳宏基,葛萌,程海波 .MyBatis 框架在数据持久层中的应用研究 [J].微型电脑应用,2023,39(01):73-75.
- [15] 于鸿瑞. 线上营业厅系统的设计与实现[D].北京交通大学,2023.
- [16] 王俊. 基于知识图谱的饮食健康知识问答系统[D].南昌大学,2023.

- [17] 曹陈涵. 面向运维工单的自动化管理系统[D].南京邮电大学,2023.
- [18] 李满. 基于微服务架构的 SaaS 应用集成框架设计与实现[D].西安电子科技大学,2021.
- [19] 范展源,罗福强.JWT 认证技术及其在 WEB 中的应用[J].数字技术与应用,2016(02):114.
- [20] 黄艳东. 基于网络评论情感分析的商品评价系统设计与实现[D].北京交通大学,2023.
- [21] 张洪春.软件测试工程化模型及应用研究[J].中央民族大学学报(自然科学版),2016,25(01):40.

## 致 谢

四年的大学生活就快走进尾声，我们的校园时光就要划上句号，心中是无尽的难舍与眷恋。从这里走出，对我的人生来说，将是踏上一个新的征程。

回首四年，生活中有快乐也有艰辛。感谢四年来的任课老师和辅导员对我孜孜不倦的教诲，对我成长的关心和爱护。衷心感谢陈子军教授对本人毕业设计的指导，老师认真负责的态度和严谨的治学精神使我受益匪浅，终生受益。

学友情深，情同兄妹。四年里，感谢班级同学和宿舍舍友对我的关心和帮助，我们一同走过，充满了关爱，给我留下了值得珍藏的最美好的记忆。

在我的十几年的求学历程中，离不开父母的鼓励和支持，是他们辛勤的劳作，无私的付出，为我创造了良好的学习条件，感谢父母对我的抚养与培育。

我会在之后的人生旅程中，用知识武装自己的头脑，争做合格的社会主义接班人，回报父母，回馈祖国，用一腔热血投身到社会主义的建设中去。



## 附录 1 开题报告

### 一、综述本课题国内外研究动态，说明选题的依据和意义

#### (一) 国内外研究现状

##### 1、知识图谱的研究现状

随着人工智能相关技术的不断进步和发展，知识图谱（Knowledge Graph）逐渐成为人工智能的重要分支，知识工程在大数据环境中的成功应用。知识图谱的本质可以看做是语义网络（Semantic Network）的知识库。语义网络借助图解够对信息进行结构化存储，在 20 世纪 50 年代由 M. Ross Quillian 和 Robert F. Simmons 等人提出。知识图谱在语义网络的基础上，保留了语义网络中使用图的结构对信息进行表示的特点，并以结构化的形式对数据涉及的实体和实体间的关系进行标识。知识图谱通常以三元组的形式对实体间关系进行标识，可以抽象为  $G = (E, R, S)$ ，其中  $E = \{e_1, e_2, \dots, e_{|E|}\}$  是代表实体的集合，包含实体的个数是  $|E|$  个；同样， $R = \{r_1, r_2, \dots, r_{|R|}\}$  对关系集合进行描述； $S \subseteq E \times R \times E$  则对图谱中所有三元组进行表征。由此，知识图谱以实体为基本单位，通过关系将实体连接以形成知识网络。

近年来，在构建知识图谱方面踊跃出大量的研究成功。随着人们需求不断增加以及自然语言处理技术的发展，大量知识图谱相关项目开始落地实施。从知识图谱构建所依赖的技术上区分，可以分为两类：一类是以 WordNet, HowNet 等知识库为代表的早期以手工构建的本体库；另一类则是以 YAGO, DBPedia 等为代表的借助自动构建技术且面向开放领域的知识图谱。从图谱面向的范围和领域区分，可以分为面向通用领域知识图谱与面向垂直领域知识图谱两类。面向通用领域的知识图谱一般应用于搜索等场景，要求包含知识的全面性，但对知识的准确性的要求相对较低，比较有代表性的通用图谱有国内的 Zhishi.me、CN-DBpedia、搜狗知立方和百度知心以及国外的 Freebase、YAGO、DBPedia 等。面向垂直领域的知识图谱主要应用于智能问答、知识推理以及辅助决策等要求知识准确性场景，目前垂直领域的知识图谱主要涉及旅游、交通和医学等方面。

国外方面，关于饮食推荐的研究起步较早，随着 1995 年 Epicurious 食谱网站建

成，1997 年 Food, Allrecipes 等食谱网站相继建成，大量关于食谱推荐系统的研究开始涌现。2010 年，美国麻省理工学院计算机科学与人工智能实验室的研究人员和哈佛大学公共卫生学院的营养学家共同合作创建了 Recipe M+<sup>[1]</sup> 公开数据库。文献[2]通过整合食材，宗教、营养、喜好等多个领域的知识来建立数据库，对所有食品进行检索，然后根据用户的身体健康状况向他们推荐适合的食品。

目前，知识图谱的构建主要通过存在于互联网中多种形式的数据作为数据源，包括结构化数据、半结构化数据和非结构化数据，进行知识抽取，经过一系列的数据处理选择合适的存储形式建立知识图谱。根据面向领域以及数据形式的不同会采用不同的构建流程，目前主流的图谱构建流程包括自顶向下（top-down）和自底向上（button-up）两种模式。其共有的核心流程主要可以分为知识抽取、融合以及存储三个步骤。目前在健康饮食领域，多数知识图谱构建工作集中在症状、疾病、药物等医疗主题，少有将食物与健康进行关联，为日常饮食服务。

## 2、推荐系统的研究现状

伴随着推荐算法的发展，传统的推荐算法，如协同过滤算法、以及基于内容的推荐算法由于数据极度稀疏而无法满足人们的需求。随着互联网的发展，越来越多的蕴含用户行为信息的数据产生并被获取，包括图像、文本以及标签等多元异构信息<sup>[3]</sup>。混合推荐算法通过利用这些辅助信息，可以有效地缓解传统推荐算法中数据稀疏与冷启动问题。

知识图谱中蕴含丰富的实体与实体之间的关系，能够在一定程度上解决传统推荐算法所带来的难题。基于知识图谱的推荐模型大多以现有的推荐模型为基础，通过知识图谱中将项目、用户等实体的结构化知识描述得更加细粒度化，利用知识推理得出更加深层次的信息，丰富推荐结果的多样性。文献[2]通过从美食杰网站获取用户信息数据，构建了用户-食谱偏好数据集，进行实体抽取，提出了结合知识图谱的个性化推荐模型。文献[3]构建了基于知识图谱的 RippleNet 推荐模型，通过分析知识图谱中节点和语义关系获得节点间深层次的潜在关系，并结合物品推荐思想在知识图谱中获得用户的潜在偏好信息。

### （二）选题的依据和意义

随着中国经济的高速发展，中国居民的生活水平大幅提升，家庭生活 and 社交生活中，饮食都扮演着日益重要的角色。饮食健康、营养均衡已经成为人们越来越关注的问题，对饮食个性化、多样化的需求也随之大幅提升。“双循环”经济发展格局是党

中央的战略性决策，人民日益增长的美好饮食需求和行业发展不平衡、不充分之间的矛盾阻碍了经济的发展，也揭示了食品消费市场的巨大潜力。个性化的饮食推荐系统还能刺激人们的食品消费，促进食品行业、农业国内大循环，国内国际双循环，为中国经济增长助力。

传统的搜索引擎只考虑了网页与搜索关键词的相关度，帮助用户过滤和筛选信息，在一定程度上，降低了获取饮食信息的难度，但没有提供个性化的服务，仍然需要花费大量时间搜索和筛选以获得适合个人的饮食菜谱。相较于传统的搜索，推荐系统既能解决信息过载问题还能满足人们的个性化、多样化的饮食需求。

在众多外部信息中，知识图谱作为一个由语义网发展而来的知识库，包含了大量的领域实体和实体之间关系的描述，具有丰富的语义信息。近年来，知识图谱也已成为语义搜索、知识问答等人物的主要知识来源之一，取得良好的表现。知识图谱中包含了多种类型的实体和实体间的语义关系，因此将知识图谱引入到个性化推荐中，能够缓解物品冷启动问题、发现推荐物品之间的潜在关联关系、提升推荐的准确性和多样性。

基于以上原因，设计实现一个结合知识图谱的饮食推荐系统，加快人们对日常个性化、多样化的饮食的获取效率，刺激人们的食品消费需求，为食品产业升级创新提供支持。

## 二、研究的基本内容，拟解决的主要问题

基于上述研究目标，本项目的研究内容主要有以下几个方面：

- 1、整合营养学、食材特性、烹饪方法等多维度信息，进行数据建模，建立食物、营养元素、烹饪关系等节点，进行实体抽取、关系抽取和属性抽取，再进行存储工作，构建饮食知识图谱（三元组）。
- 2、构建用户-饮食偏好数据集，并对知识图谱数据进行预处理。本项目需从美食杰等网站获取数据，建立用户交互记录，即用户-饮食偏好数据集。
- 3、寻找合适的数据存储形式。饮食知识图谱可以使用图数据库进行存储，如 Neo4j；用户-饮食偏好数据集可以使用 MySQL 数据库进行存储。
- 4、以构建的数据集为基础，实现结合知识图谱的推荐模型，完成用户的个性化饮食推荐结果生成，并能定期更新数据，继续训练模型并使用。
- 5、设计和实现 B/S 架构的 Web 系统，包括用户登录、注册、查询和浏览饮食信

息等基本功能。系统可以根据知识图谱中饮食的类别，生成一个类别列表，当完成注册时，跳转到喜好类别标记页面，让用户选择自己喜欢的类别，方便后续进行饮食个性化推荐。当用户在系统中浏览饮食详情、为饮食点赞时，系统会记录下用户的行为，并将行为统一格式存入数据库中。

6、结合上述 Web 系统，完成推荐模型的调用，为用户提供个性化推荐服务。

7、设计和实现社区交流模块，使用户可以畅所欲言，分享日常饮食、注意事项等内容。用户可以进行发布主题、回复主题和点赞信息等基本操作。

8、设计和实现管理员后台系统，实现管理用户、饮食信息、模型训练、社区交流等内容。系统开放使用后，新产生的用户反馈数据会定期进行模型训练，更新模型的参数，并将其保存到数据库中。

### （三）拟解决的主要问题

#### 1、数据集获取和构造问题

本系统需要饮食知识图谱和用户-饮食偏好数据两大数据集。饮食数据需要从各大美食平台进行爬取数据，一部分是用户交互记录，另一部分是通过饮食详细信息构造的饮食知识图谱三元组文件。其中，饮食知识图谱三元组文件需要进行实体和关系的抽取。为了构造一个标准化数据集，首先对推荐算法公用数据集 ml-1m 进行分析，然后将爬取的食谱信息进行数据分析处理使其各项分布基本与 ml-1m 保持一致，便于后续实验进行对照。其次，将饮食知识图谱存入 Neo4j 图数据库，将用户-饮食偏好数据集存入 MySQL 数据库。

#### 2、模型训练问题

推荐模型是整个推荐系统的核心部分，模型的效果直接决定了推荐系统的价值，结合推荐物品的特点来设计推荐模型是非常重要的。模型训练主要使用框架 Pytorch。它提供了非常全面的 API，包括基本的向量矩阵计算、各种优化算法、各种卷积神经网络和循环神经网络基本单元的实现以及可视化的辅助工具。

#### 3、数据库的设计和建立

本系统需要存储各种信息，如饮食信息、用户-饮食偏好数据、用户交互信息、用户和管理员个人信息、模型信息、推荐信息和社区交流信息等。所以，合理设计数据库表可以为后续的功能实现奠定基础。

#### 4、Web 系统设计和实现

本系统是 B/S 架构的 Web 系统，需要实现用户登录、注册、修改个人信息等基

本功能，需要能够远程调用模型，返回个性化推荐结果。在完成这些基础功能的同时，需要合理分配网站的布局，把握好网站所要体现的主要内容，完成界面设计和简化用户操作流程，给用户良好的使用体验。

### 5、设计实现交流社区

在此问题中，合理设计数据库表进行存储社区交流信息是非常重要的。交流社区涉及到主题、一级评论、二级评论等信息，合理设计存储方式可以减小开发难度。

## 三、研究步骤、方法及措施

### (一) 研究步骤

1、进行选题研究，对基于知识图谱的饮食推荐系统的操作性、创新型、使用性和可行性等角度进行分析，确定选题的实施方向和意义，最终确定论文题目。

2、进行用户的需求调查和分析，认识用户主要的需求功能，了解饮食推荐和交流方面的需求，了解饮食营养元素、食材特性、烹饪方法等多维度信息对人体健康的作用和影响，初步确定数据的组织形式和存储方法，保证功能的完善与有效。

3、进行总体设计，绘制系统架构图、功能模块图和功能流程图。其次，基于功能层次，对系统功能模块进行初步划分，保证系统的完整性。进行数据存储形式的设计，学习和使用 Neo4j 图数据库的使用方法，完成数据的存储。

4、进行数据的获取，由于目前没有一个完整的饮食开源数据库，所以需要通过爬虫技术，获取美食杰等饮食网站信息，对数据进行整合，提取知识要素，最终得到较为完整的知识图谱数据。

5、通过文献查询，了解知识图谱饮食推荐领域中所设计的模型，制定出本系统的推荐模型，并验证模型，分析评价指标。

6、进行详细设计与编码实现，通过系统模块图，分模块绘制程序流程图并编写代码完成功能。最后进行系统的试运行和测试，按用例进行多次测试，保证系统的安全和正常运行。

### (二) 方法及措施

在项目实施过程中，将采用实践调研法、文献研究法和经验总结法，完成本项目的开展工作。本项目拟定实现一个 Web 形式的饮食推荐系统。有两种实现方式：

1、方案 1：使用 JSP 动态的网页技术，其中既可以定义 HTML、JavaScript、CSS 等静态内容，还可以定义 Java 代码的动态内容，以 MySQL 作为后台数据库。

2、方案 2：使用 SpringBoot、Mybatis、Vue.js 技术进行开发，以 MySQL 作为后台数据库。根据主流的 Web 开发框架 Spring MVC 分层设计的思想，系统分为视图层（View）、控制层（Controller）以及模型层（Model）。视图层负责展示饮食数据、与用户进行交互。控制层负责分发视图层的请求，调用模型层与数据库进行交互，并将结果数据返回给视图层。模型层负责与数据进行交互，封装并实现具体的功能。相应地，从系统代码的角度来看，又可以分为交互层、业务逻辑层及数据层。视图层对应交互层，模型层被分为数据层与业务逻辑层，控制层位于业务层和交互层之间。

在方案 1 中，JSP 技术可以在网页中编写 Java 代码，实现业务逻辑，但是模块耦合度较强，独立性较差，后续维护和修改的难度较大。在方案 2 中，前后端分离可以并行开发，提高开发效率，代码耦合度降低，代码够更清晰，易于维护和升级。前后端分离后，可以更加灵活地进行系统扩展和升级。所以项目选择方案 2 实现。

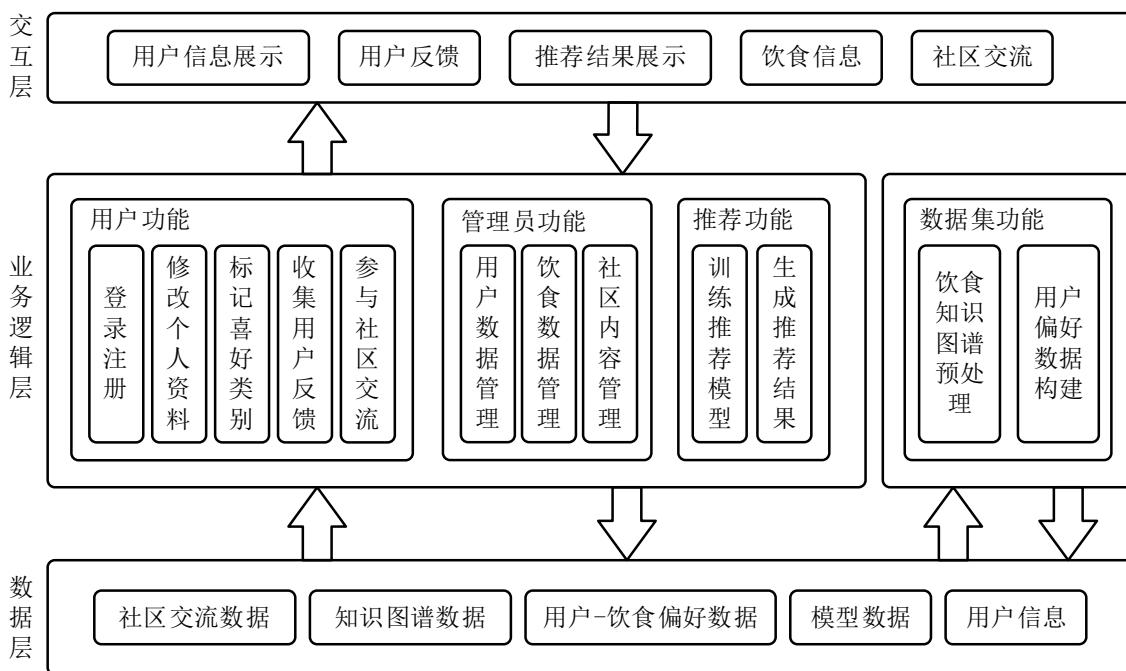


图 1 系统架构图

系统的整体架构如图 1 所示，数据层用户存储知识图谱数据、用户-饮食交互数据、用户个人信息、社区交流数据等信息。知识图谱数据是指项目所使用的饮食知识图谱。用户-饮食偏好数据包括两部分，一部分是通过爬虫所获取用于模型训练的用户-饮食偏好数据，另一部分是系统正常使用后，用户在系统中产生的反馈数据。

如图 2 所示，为推荐系统的功能用例图。个性化推荐系统的目的是为用户服务，

用户是系统的核心，因此用户相关的登录、注册、显示和修改个人信息等功能在系统中不可或缺。为了能够给新注册到系统用户提供更快、更精准的推荐服务，需要让用户选择一些喜欢的类别。在系统正常使用的过程中，用户反馈数据不断改善推荐给系统内容，以提供更精准的推荐服务。

在用户一侧，可以实现登录注册、标记喜好类别、修改个人资料和密码、与饮食信息进行交互，从而生成交互信息。同时，系统也可以根据用户喜好类别进行推荐饮食结果和根据用户交互信息来获取推荐类别。用户也可以参与社区交流，发布主题，回复内容，和对自身所发布主题进行管理。

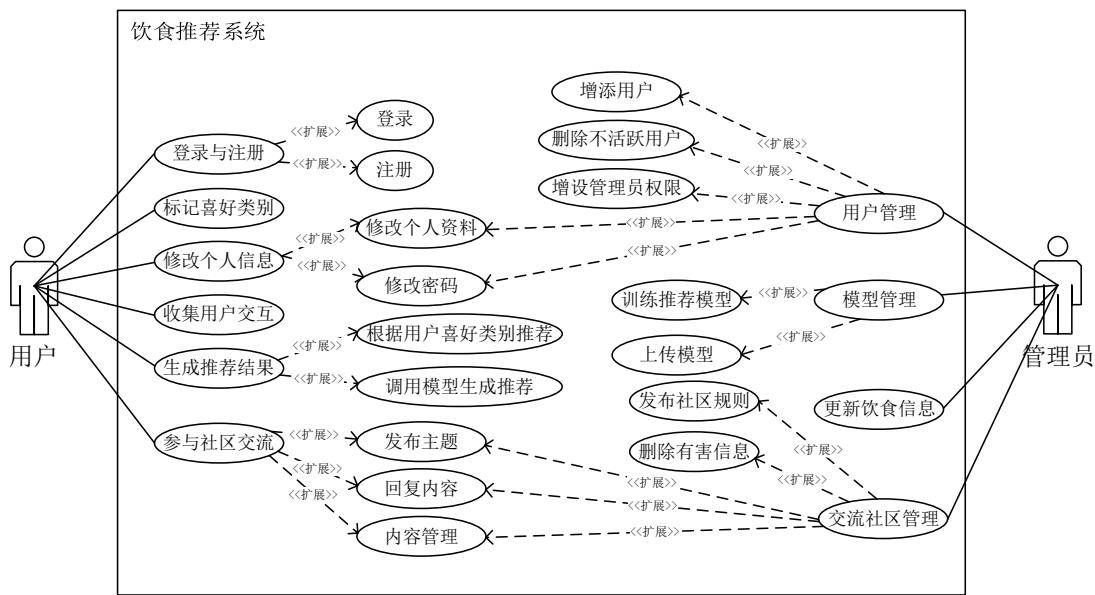


图 2 系统功能用例图

在管理员一侧中，其中包含用户管理、模型管理、更新饮食信息、交流社区管理。在用户管理中，管理员可以增添用户，删除不活跃的用户，增设和分配管理员权限，也可以修改个人资料和密码。在模型管理中，管理员可以定期训练推荐模型和上传模型以供系统使用。管理员也可以紧随时代潮流更新热门饮食信息。同时，管理员可以对交流社区进行管理，能够发布社区规则，删除有害信息，发布主题，回复内容和内容管理等功能。

如图 3、图 4 所示，为用户端和管理员端系统功能模块图，其中包括用户模块、模型训练模块、管理员模块。在用户模块中，用户可以登录和注册信息、标记自己的喜好类别、浏览相关的用户信息、修改个人信息，并且参与社区交流。在模型训练模块中，可以进行数据集获取和预处理，进行模型训练和部署。在管理员模块中，管理

员可以对用户和权限进行管理，用户交互的管理，饮食数据的管理等。

推荐模块包含饮食推荐模型的训练于推荐结果的生成两部分需求。训练推荐模型主要是指系统投入使用后在系统新收集的用户反馈数据上定期进行训练维护。用户在使用系统的过程中，也会产生一定量的交互数据。因此，需要定期将产生的数据进行训练，不断学习新数据特征，用于以后的推荐。当系统中用户发出推荐请求时，对于系统中的新用户，系统应根据用户标记的喜好类别进行推荐；对于系统中已有的用户，系统应调用最新的模型进行预测，模型预测结果即为用户推荐的饮食，并将推荐的结果显示给用户。

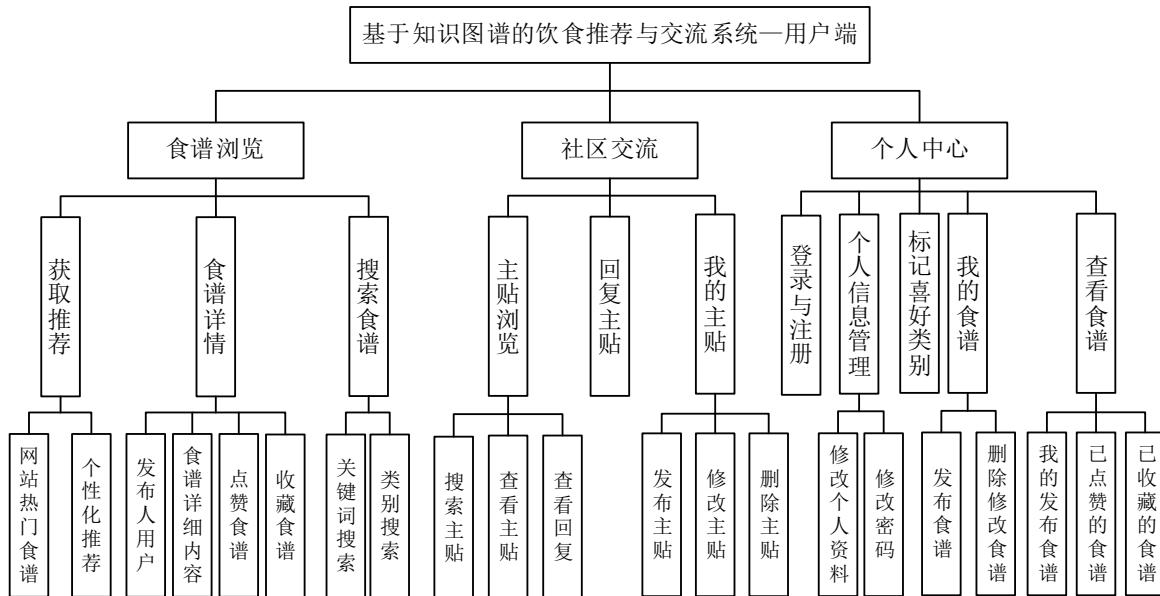


图 3 用户端系统功能模块图

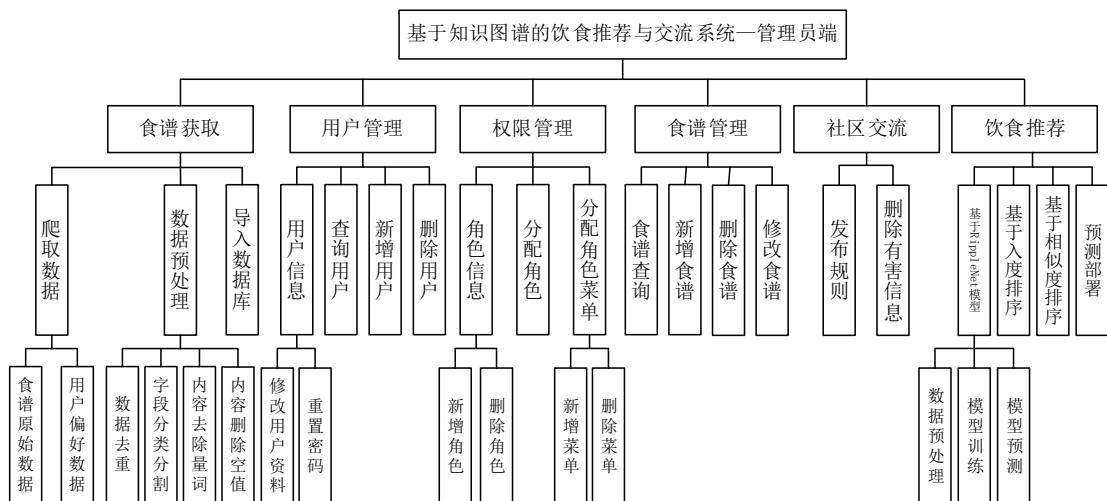


图 4 管理员端系统功能模块图

#### 四、研究工作进度

##### 1、准备与启动阶段（12月12日—寒假）

查阅相关文献，确定选题内容，针对选题的具体内容进行文献资料查阅与数据收集整理等工作，着手准备开题报告等相关工作。

##### 2、开题阶段（2024年2月26日—3月17日，第1-3周）

对已搜集的资料加以整理，论证选题的可行性和实际性。按照毕设选题进行调研分析和方案设计，拟定具体的研究内容、计划和路线，补充理论知识，撰写开题报告。

##### 3、中期阶段（2024年3月19日—4月28日，第4-9周）

针对开题考核中存在的问题加以整改和完善，开始进行选题的总体设计和详细设计，完成饮食知识图谱、用户-饮食偏好数据集的获取和整理工作，进行推荐系统模型训练，加以验证和改善，完成原型系统的初步设计和开发工作，以验证实施方案和预期结果，完成毕业设计论文的初稿。

##### 4、完善阶段（2024年4月29日—5月26日，第10-13周）

根据中期考核中存在的问题及时改进，完善系统的功能，并对系统进行完整测试，继续收集整理训练数据集，进行模型训练。进一步充实完善毕业论文初稿的各项内容。

##### 5、答辩阶段（2024年5月27日-6月9日，第14-15周）

进行毕业论文完善和定稿，依据毕业论文和系统制作答辩PPT。

#### 五、主要参考文献

- [1] Marin Javier;;Biswas Aritro;;Ofli Ferda;;Hynes Nicholas;;Salvador Amaia;Aytar Yusuf;;Weber Ingmar;;Torralba Antonio.Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images.[J].IEEE transactions on pattern analysis and machine intelligence,2019.
- [2] Health, Food and User's Profile Ontologies for Personalized Information Retrieval[J]. Tarek Helmy;;Ahmed Al-Nazer;;Saeed Al-Bukhitan;;Ali Iqbal.Procedia Computer Science,2015.
- [3] 江山. 基于知识图谱的个性化食谱推荐算法研究与实现[D].浙江理工大学,2023.
- [4] 仲亮靓. 基于饮食知识图谱的食谱推荐系统的设计与实现[D].东南大学,2020.
- [5] 董洪伟. 基于知识图谱的菜品推荐系统[D].北京林业大学,2021.

- [6] 刘伟江,郝一哲.融合知识图谱的 RippleNet 推荐模型优化研究[J].现代情报,2021(09).
- [7] 漆桂林,高桓,吴天星.知识图谱研究进展[J].情报工程,2017,3(01):4-25.

## 附录 2 文献综述

### 一、课题国内外现状

近年来，在构建知识图谱方面踊跃出大量的研究成功。随着人们需求不断增加以及自然语言处理技术的发展，大量知识图谱相关项目开始落地实施。从知识图谱构建所依赖的技术上区分，可以分为两类：一类是以 WordNet, HowNet 等知识库为代表的早期以手工构建的本体库；另一类则是以 YAGO, DBpedia 等为代表的借助自动构建技术且面向开放领域的知识图谱。从图谱面向的范围和领域区分，可以分为面向通用领域知识图谱与面向垂直领域知识图谱两类。面向通用领域的知识图谱一般应用于搜索等场景，要求包含知识的全面性，但对知识的准确性的要求相对较低，比较有代表性的通用图谱有国内的 Zhishi.me、CN-DBpedia、搜狗知立方和百度知心以及国外的 Freebase、YAGO、DBpedia 等。面向垂直领域的知识图谱主要应用于智能问答、知识推理以及辅助决策等要求知识准确性场景，目前垂直领域的知识图谱主要涉及旅游、交通和医学等方面。

国外方面，关于饮食推荐的研究起步较早，随着 1995 年 Epicurious 食谱网站建成，1997 年 Food, Allrecipes 等食谱网站相继建成，大量关于食谱推荐系统的研究开始涌现。2010 年，美国麻省理工学院计算机科学与人工智能实验室的研究人员和哈佛大学公共卫生学院的营养学家共同合作创建了 Recipe M+<sup>[1]</sup>公开数据库。文献[2]通过整合食材，宗教、营养、喜好等多个领域的知识来建立数据库，对所有食品进行检索，然后根据用户的身体健康状况向他们推荐适合的食品。

目前，知识图谱的构建主要通过存在于互联网中多种形式的数据作为数据源，包括结构化数据、半结构化数据和非结构化数据，进行知识抽取，经过一系列的数据处理选择合适的存储形式建立知识图谱。根据面向领域以及数据形式的不同会采用不同的构建流程，目前主流的图谱构建流程包括自顶向下（top-down）和自底向上（button-up）两种模式。其共有的核心流程主要可以分为知识抽取、融合以及存储三个步骤。

### 二、研究主要成果

文献[3]使用 Recipe1M+ 数据集进行了基于内容的推荐算法和协同过滤推荐算法

的研究，实现了能够发现相似用户的偏好食谱和与用户历史操作记录相似的偏好食谱的功能。文献[4]利用 Recipe1M+数据集构建了饮食知识图谱，并开源了相关构建知识图谱的代码和数据集，为其他研究人员进行知识图谱的推荐和问答提供了理论和数据基础。文献[5]通过爬取 Allrecipe 网站数据，构建了 MealRec 饮食推荐数据集，该数据集包含了五百多位用户、七千两百多份食谱和三百八十多份餐点，并提出了一种基于类别约束的饮食推荐模型，通过实验测试和对比验证了数据集和模型的有效性和准确性。

国内方面，由于中国的食谱菜系众多，烹饪方式繁杂，食谱的应用专业性更强。国内的食谱和饮食相关的推荐系统研究还比较初步。中华美食网（2003 年建立）、美食杰（2007 年建立）、豆果网（2008 年建立）等食谱网站的兴起，国内食谱推荐系统的研究开始有了数据来源。文献[6]提出了 RippleNet-CA 模型，利用共有属性采样算法，加强了用户与食谱间的历史交互记录数据，改善了用户与食谱交互不充分的问题，RippleNet 模型的预测效果有了显著增强。文献[7]使用爬虫技术爬取美食杰网站数据信息，构建了食谱数据集，并引入了三元组上下文、实体描述文本的知识表示学习方法与贝叶斯个性化排序方法进行模型训练的联合学习方法，能够通过用户的历史兴趣预测用户的偏好，实现基于知识图谱的个性化食谱推荐。国内的饮食推荐领域相较于国外的推荐领域数据集广泛，信息多样化，研究难度大，更具有研究价值。

### 三、发展趋势

国外的食谱推荐研究更加成熟和体系化，有很多公开的数据集和开源代码仓库，在现实落地应用上也广泛。国内的食谱推荐研究相较于国外数据集信息更加多样、复杂，研究难度更大，更加具有研究价值。此外，还有不少食谱推荐研究涉及到营养配餐、多模态食谱推荐、食谱生成等领域，为食谱推荐研究提供了更多思路。

### 四、存在问题

国内食谱种类繁多，信息量大，研究难度大。存在用户与食谱交互记录获取困难，用户交互不充分，数据稀疏等问题。

### 五、主要参考文献

- [1] Marín J, Biswas A, O li F, et al. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and ood images[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(1): 187-203.
- [2] Health, Food and User's Profile Ontologies for Personalized Information Retrieval[J]. Tarek

## 附录 2 文献综述

---

---

- Helmy;;Ahmed Al-Nazer;;Saeed Al-Bukhitan;;Ali Iqbal.Procedia Computer Science,2015.
- [3] Bushra N, Hasan M. uicklyCook: A User-Friendly Recipe Recommender[C]//Machine Learning and Metaheuristics Algorithms, and Applications: First Symposium, SoMMA 2019,Trivandrum, India, December 18–21, 2019, Revised Selected Papers 1. Springer Singapore,2020: 245-254.
- [4] Haussmann S, Seneviratne O, Chen Y, et al. FoodKG: a semantics-driven knowledge graph for food recommendation[C]//The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18.Springer International Publishing, 2019: 146-162.
- [5] Li M, Li L, Xie , et al. MealRec: A Meal Recommendation Dataset[J]. arXiv preprint arXiv:2205.12133, 2022.
- [6] 江山. 基于知识图谱的个性化食谱推荐算法研究与实现[D].浙江理工大学,2024.
- [7] 仲亮靓. 基于饮食知识图谱的食谱推荐系统的设计与实现[D]. 东南大学, 2019.



## 附录 3 中期报告

### 一、任务书中本阶段工作目标与任务要求

#### (一) 本阶段工作目标

- 1、利用爬虫框架编写爬取美食杰网站食谱数据和用户偏好数据代码，在保证安全的情况下获取美食杰网站数据。
- 2、对爬取数据进行初步清晰，删除空值、无意义的数据，统计数据数目，验证数据的准确性和真实性。
- 3、根据需求分析内容和功能进行数据库设计，保证数据库表字段的设计规范和要求，确保数据存储能够满足功能需求。
- 4、使用 SpringBoot、Mybatis 搭建后端项目，完成 MySQL 数据库、权限验证等配置工作，完成控制层向前端提供的接口。
- 5、使用 Vue、Node.js、Element Plus 搭建前端项目（用户端和管理员端），使用 router、axios 封装请求函数和过滤器。
- 6、进行模型的学习和训练，在训练过程中依照训练反馈不断修正和完善原始数据和参数，不断完善训练结果。
- 7、完成用户登录注册、修改个人信息和密码的功能，管理员对用户操作。
- 8、食谱数据导入 MySQL 数据库，完成食谱的展示、点赞、收藏等功能。

#### (二) 任务要求

- 1、在遵守网络规范的前提下，合理提取网站数据。
- 2、确保数据的合理性。
- 3、保证开发的质量和内容。
- 4、变量和文件命名要规范

### 二、目前已完成任务情况

#### (一) 功能模块设计

如图 1 所示，为本项目的用户端功能模块图。在用户端中，本项目主要包含食谱浏览、社区交流和我的三大模块。在食谱浏览中，未登录用户可以进行浏览当前网站热门食谱和食谱详细内容，并且根据关键词或类别搜索食谱。登录用户可以获取基于

个人喜好的个性化推荐。在社区交流中，用户可以搜索和查看主题及回复，并且可以对喜好的主贴进行回复，在我的主贴中，用户还可以发布主贴，并且能够对自己发布主贴的内容进行修改或删除。在个人中心中，用户可以完成登录注册、对个人信息和密码进行修改，并且可以进行发布食谱或删除修改食谱，也可以查看用户已发布、已点赞或已收藏的食谱。

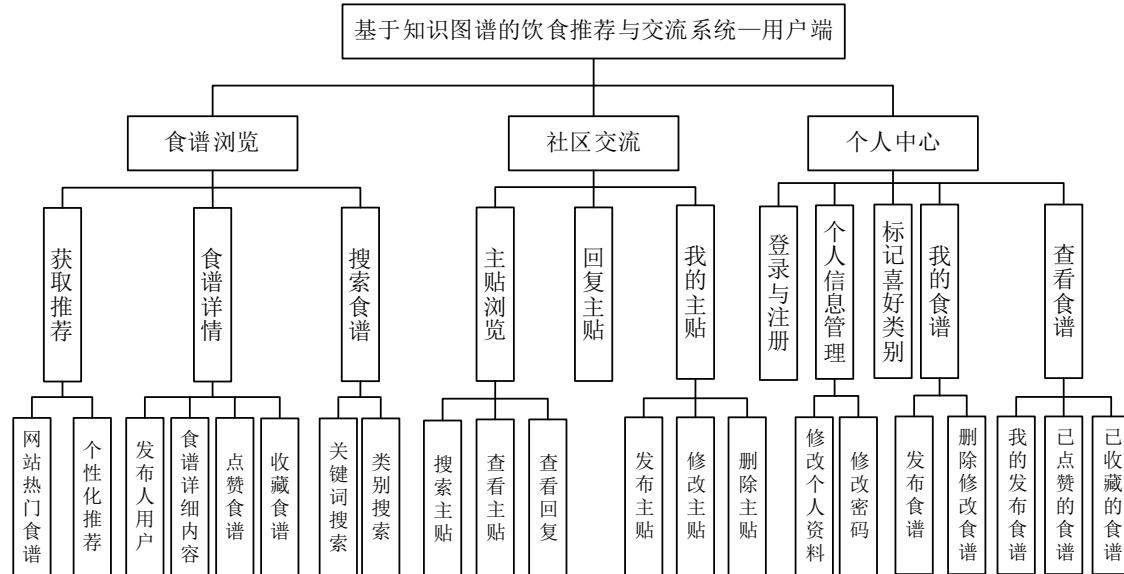


图 1 用户端功能模块图

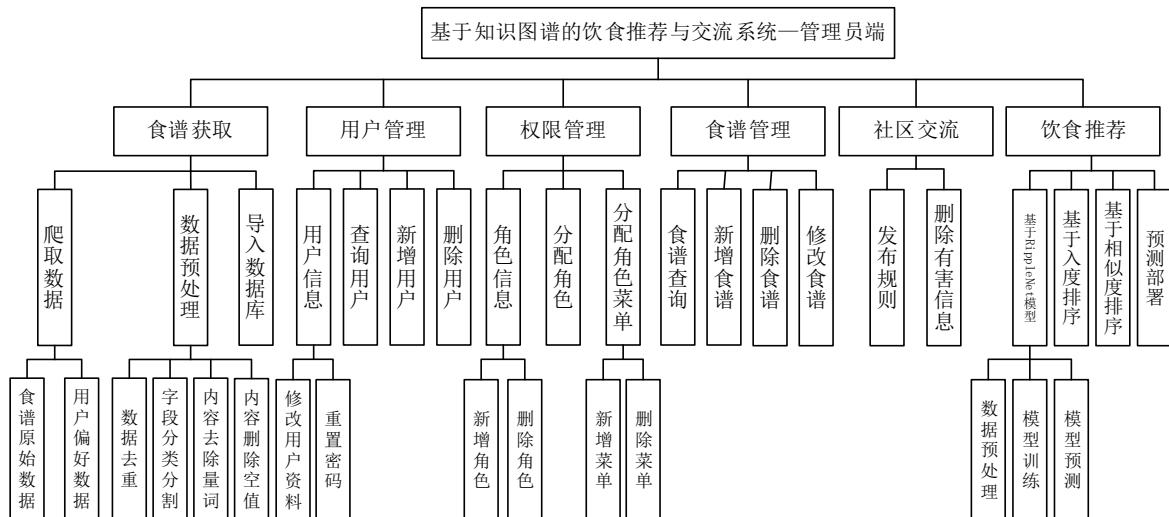


图 2 管理员端功能模块图

如图 2 所示，为本项目的管理员端的功能模块图。在管理员端，其中主要包含食谱获取、用户管理、权限管理、食谱管理、社区交流和饮食推荐。在食谱获取中，本项目从美食杰网站爬取食谱原始数据和用户偏好数据，并对原始数据进行去重、分割、去掉空值等处理获得完整数据，并将食谱数据转化为 Excel 表格形式导入数据库

中，为后续项目使用提供基础。在用户管理中，管理员可以对用户的资料信息进行修改，重置用户的密码。管理员可以新增、查询和删除用户账号。

本项目的权限管理主要是针对管理员对不同的菜单进行分权限管理，例如普通管理员可以对食谱信息进行修改，而高级管理员既可以对食谱信息进行修改还可以对用户账号进行相应操作，这样可以在一定程度上保证系统的数据稳定和安全。在权限管理模块中，具有权限管理的许可的管理员可以新增角色或删除角色，也可以对管理员的角色进行分配，对角色所具有的菜单进行分配管理。在食谱管理中，管理员可以对食谱进行一些增删改查的操作。在社区交流中，管理员可以发布相关规则或删除有害信息（主贴或回复贴）。本项目在饮食推荐上，使用基于 RippleNet 模型、基于入度排序、基于相似度排序进行推荐，并将预测程序使用 Python 进行部署。

## （二）获取数据

本文是基于知识图谱的饮食推荐系统，因此本文数据集方面包含饮食知识图谱数据和用户-食谱偏好数据集的需求。爬取数据使用的是 scrapy 框架，采取可读性更强的 xpath 代替正则表达式，可同时在不同的 url 上爬取数据，并可以通过管道的方式进行存储。

在调研美食杰网站数据分布时，有两种方案爬取数据。方案一是根据美食杰网站所列举的大类和小类为主要切入点爬取每个类别下的食谱信息。方案二是直接爬取用户的收藏列表中的食谱数据，既可以获取食谱数据，也可以获得用户-食谱偏好数据。

A	B
1 食谱名称	计数项：食谱名称
2 可乐鸡翅	17
3 糖醋排骨	15
4 宫保鸡丁	14
5 鱼香肉丝	13
6 酸菜鱼	12
7 水煮肉片	12
8 鱼香茄子	11
9 水煮牛肉	10
10 糖醋里脊	10
11 麻婆豆腐	10
12 红烧排骨	10
13 红烧肉	10
14 油焖大虾	9
15 清蒸鲈鱼	8
16 梅菜扣肉	8
17 蚂蚁上树	8
18 红烧茄子	8
19 红烧带鱼	8
20 洋葱炒鸡蛋	7
21 凉拌黄瓜	7
22 回锅肉	7
23 干煸四季豆	7
24 西红柿炒鸡蛋	6

图 3 重复食谱数据统计图

如图 3 所示，为方案一爬取家常菜类别的部分结果，其中出现了高达 17 次的重复食谱，同时也存在不属于家常菜类别的食谱，数据准确性较差。在测试多个类别后，仍然出现类似的情况，所以推翻了方案一。在方案二中，需要遍历所有的用户列表，再从用户的主页进入爬取所收藏的食谱信息。这些爬取到的用户收藏食谱信息通过去掉重复值即可获得食谱信息。

美食杰的用户列表页码在 URL 的 page 参数中，可以提前设置需要爬取用户的页数范围，构造出用户列表的 URL 地址数组，然后再依次进入用户主页获取收藏列表进行分析食谱信息。本文的具体数据爬取步骤如下：

- (1) 通过提前设置的页码范围构造用户列表的 URL 地址数组；
- (2) 从这些用户列表网页中解析用户主页的 URL 地址；
- (3) 爬取 (2) 中获得的所有用户主页 URL 对应的网页；
- (4) 从 (3) 中获取用户主页中的个人信息及对应收藏列表的食谱网页的 URL 地址，将其插入待爬取 URL 地址队列中；
- (5) 对 (4) 中获取的食谱网页进行解析，得到食谱数据。

由于美食杰网站的数据都是“所见即所得”的，即网页中显示的数据都是直接保存在 HTML 页码中，不需要向后台发送请求动态获取，所以直接通过 xpath 方式来获取 HTML 中渲染的标签内数据即可。如图 4 所示，为爬取数据流程图。

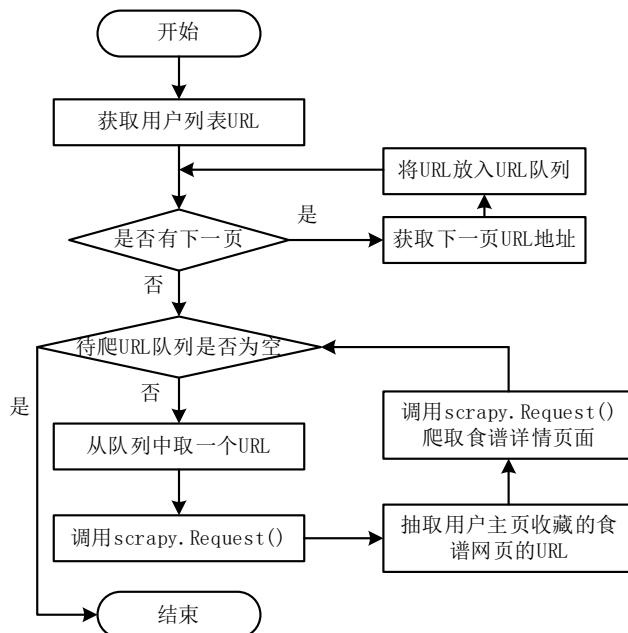


图 4 Scrapy 爬虫流程图

附录 3 中期报告

首先，先获取所有用户列表的网址集合，再依次遍历获取用户的主页网址，将其加入到待爬取 URL 队列中。访问每一位用户的主页地址，打开用户的收藏食谱界面，从第一个食谱开始依次循环取出所有的收藏食谱。

在爬虫程序运行时，会因为数据短缺等问题出现各种程序异常和中断情况，在程序中需将异常抛出，使程序能够顺利执行。为了获得高质量和较为准确的数据，在程序中设置多处判空和 NULL 的操作，并将其删除。在爬取到一条食谱交互数据后，scrapy 通过预设的格式输出添加到 Excel 表格中，最终生成原始网站数据。由于网站自身的限制和安全的考虑，数据爬取设置了较慢的爬取速度，所以在项目实现的过程中，适时开启爬虫程序，以获得更多的原始数据。

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	木樨豌豆	img	craft	家常味	鸡蛋140克 豌豆(鲜)100克 生花生油20毫升 盐 少量 麻	准备食材：鸡蛋打散	258.5	18.41	12	5分钟	初级入门	12.1,1.1	
3	南瓜小米粥	https://st.cn.me	煮	甜味	老南瓜200克 小米100克 水适量	准备老南瓜和小米	725	10.92	0.5	60分钟	新手尝试	20.0,0.0	
4	柠檬葱油焗鸡腿	https://st.cn.me	焗	葱香味	鸡腿2只 葱油2勺 蚝油1勺 海盐适量 姜片5片 蒜1头	准备好鸡腿、油、姜片、蒜	863	1.75	4.3	60分钟	新手尝试	0.0,0.0	
5	土豆烧鸡腿	https://st.cn.me	炒	家常味	鸡腿3只 土豆300克 洋葱2个 食用油适量	土豆切块，洋葱切丝，油烧热后放入土豆翻炒，再加入洋葱翻炒，最后加入鸡腿翻炒	866	7.49	2.8	15分钟	未知	0.0,0.1,1	
6	照烧鸡腿	https://st.cn.me	烧	咸鲜味	鸡腿3只	照烧汁1大勺 白砂糖1勺 盐15克 鸡翅焯水，去鸡皮	113.9	3.78	4.5	15分钟	新手尝试	0.0,1.0	
7	青椒土豆烧鸡	https://st.cn.me	烧	香辣味	鸡翅根500克 土豆3-5个	青椒切块，土豆切块，油烧热后放入土豆翻炒，再加入青椒翻炒，最后加入鸡翅根翻炒	116.7	10.15	7	60分钟	初级入门	0.0,1.2,1	
8	土豆炖茄子	https://st.cn.me	炖	酱香味	茄子2个 土豆2个	油适量 盐适量 五香粉适量 茄子在淡盐水里泡32分钟	132	4.9	0.2	10分钟	初级入门	0.0,0.0	
9	牛肉炖土豆	https://st.cn.me	炖	咸鲜味	牛腩500克 土豆2个 胡萝卜	土豆切块，胡萝卜切块，油烧热后放入土豆翻炒，再加入胡萝卜翻炒，最后加入牛腩翻炒	143	8.61	0.2	90分钟	初级入门	1.0,0.1	
10	新奥尔良烤翅	https://st.cn.me	烤	香辣味	鸡翅中500克	盐2匙 芝麻油1汤匙 清水100ml 鸡翅洗净沥干水，加入盐、芝麻油、清水，搅拌均匀，腌制12-24小时	122.4	3.85	6.4	15分钟	初级入门	1.0,0.0	
11	苹果红枣红糖水	https://st.cn.me	煨	甜味	苹果一个 红枣适量 枸杞半斤	将半斤红枣和一斤枸杞洗净，加水煮开，加入苹果和红枣，小火慢炖，直到苹果软烂即可	65.7	11.27	0.2	2小时	初级入门	0.2,0.0	
12	涼拌香辣海帶絲	https://st.cn.me	拌	香辣味	海带丝一份	小米辣六七根 [根据自己今天给你们看的量572g]	6.7	6.44	21	30分钟	初中水平	10.1,0.1	
13	红焖猪蹄	https://st.cn.me	炒	家常味	猪蹄	葱 姜 白芷2片 桂皮1块 猪蹄焯水，放入料酒156ml	156	6	11.3	5分钟	新手尝试	11.1,0.0	
14	红枣糕	https://st.cn.me	烤	甜味	红枣100克	糖80克 玉米油100克 小红枣10个 下锅煮，349.1ml	349.1	24.36	20.8	30分钟	未知	12.2,1.1	
15	电饭锅版排骨焖饭	https://st.cn.me	焖	家常味	排骨300克 红萝卜半根 干贝油2勺 生抽2勺 姜丝	米饭扣肉图@https://173.com	133	5.18	12.4	60分钟	初中水平	10.1,0.0	
16	杭椒牛柳	https://st.cn.me	炒	黑椒味	牛里脊肉200克 杭椒一根	杭椒切段，牛里脊肉切片，油烧至15毫升，生抽10毫升，准备牛里脊肉200克	101.7	5.25	2.6	10分钟	初中水平	0.1,0.1	
17	杭椒牛柳	https://st.cn.me	炒	家常味	嫩牛肉200克 杭椒200克	油适量 糖适量 生抽适量 做好准备用料，700ml	101.7	2.38	2.2	5分钟	新手尝试	0.0,0.0	
18	山楂酱	https://st.cn.me	煮	酸甜味	山楂200克	柠檬汁5毫升 水适量 水冰糖 水冰糖洗净，可先剪碎	175	13.37	0.5	30分钟	新手尝试	2.0,0.0	
19	什锦蛋炒饭	https://st.cn.me	炒	咸鲜味	米饭300克 鸡蛋2个	培根适量 洋葱适量 胡萝卜 提前把饭煮熟，1020ml	120.7	11.48	3.5	15分钟	新手尝试	12.0,0.0	
20	红豆酥	https://st.cn.me	烘焙	甜味	豌豆10个 红豆200克	白糖适量 想吃好吃的红豆182g	182.9	14.49	5.6	30分钟	新手尝试	0.1,2.0	
21	菠菜蒸蛋	https://st.cn.me	蒸	家常味	鸡蛋3个 菠菜1棵	盐1克 生抽2勺 蒜泥适量 菠菜洗净去根，784ml	784	0.84	5.6	10分钟	新手尝试	0.0,0.0	
22	油爆皮皮虾	https://st.cn.me	炒	家常味	皮皮虾650克 生姜2片 蒜	盐适量 鸡精少许 皮皮虾清洗干净，15.6ml	650	0.77	0.3	5分钟	新手尝试	0.0,0.0	
23	羊肉粉丝汤	https://st.cn.me	煮	家常味	冻羊肉片200克 奇士裹粉55克 盐适量 生抽适量	羊肉粉丝汤制作方法，油烧热后放入冻羊肉片，再加入粉丝、奇士裹粉、盐、生抽，煮沸即可	73	6.44	2.4	15分钟	新手尝试	0.1,0.1	
24	板栗烧排骨	https://st.cn.me	烧	家常味	排骨400克 板栗200克	油20克 糖10克 生抽1勺 备好食材@https://161.2	161.2	7.42	10.4	60分钟	初级入门	1.1,0.1	
25	南瓜饼	https://st.cn.me	炒	家常味	南瓜300克 糯米粉300克	白糖1勺 面包糠 300克的南瓜切成113g	113.9	17.43	0.3	5分钟	新手尝试	2.1,0.0	
26	芝士焗饭	https://st.cn.me	烤	酸甜味	米饭1碗 肉末50克	番茄酱20克 鸡蛋1个 马铃薯泥一碟。番茄酱152.4ml	152.4	13.58	6	10分钟	新手尝试	0.1,2.0	
27	豆腐炒白菜	https://st.lig.mei	炒	咸鲜味	毛豆腐250g 白菜120g	食盐适量 酱油适量 葱适量 豆腐切块@http://59.9	59.9	2.45	2.6	未知	未知	0.0,0.0	
28	鮓汁油菜香菇	https://st.lig.mei	其它	其它口味	油菜100g 香菇100g	食盐适量 鲣鱼精适量 鲣油 料油、油菜、香菇5g	59.9	0.84	0.1	未知	未知	0.0,0.0	

图 5 部分网站原始数据结果图

### (三) 数据预处理

如图 5 所示，为成功爬取的食谱数据。其中，数据有很多字段包含多个值，均以 @、| 等特殊符号进行间隔，方便后续的数据操作和分割。对于“鸡蛋 140 克”、“醋适量”等字段内容，使用正则表达式进行剔除无关信息，保留“鸡蛋”等。

表 1 原始数据数量表

食谱实体数量	属性实体数量	关系数量	用户数量	用户-食谱交互数量
5708	5104	80734	1926	55367

#### (四) 构建知识图谱

为了减少容错几率，在构建知识图谱时，先用一小部分数据编写构建知识图谱程序，并提高方便接口，能够实现将相同格式文件输入到程序中并可得到最终的知识图谱，提高模块的可利用率。

本项目使用了 Neo4j 图数据库存储知识图谱，其中所涉及的工具有 Pycharm、Neo4j Desktop 和 py2neo。食谱主要的信息包括以下几个字段，食谱 id，食谱名，主食材，辅食材，烹饪方式，烹饪时间，烹饪步骤，图片等内容。本项目先将处理好后的食谱信息存入到 Excel 表格中，然后通过 Python 程序以行为单位依次读取每一条食谱信息，将食谱信息各个字段拆分为关系存储到列表中，关系有食谱-主食材，食谱-辅食材等。其次，将所有的头节点和尾节点以 Set 的集合形式进行存储，避免出现重复的实体。最后，将这些列表数据依次插入到图数据库中。

在知识图谱中，设计了两种类型的节点：食谱（recipe）节点和食谱属性节点。食谱节点个数和爬取的食谱数一致，食谱属性节点分为 6 类：主料（main\_ingredients），辅料（supplementary\_ingredients），口味（taste），烹饪方式（craft），烹饪时间（need\_time），难度级别（level）。设计了 6 种关系类型：食谱的主料（recipe\_maingredients），食谱的辅料（recipe\_fingredients），食谱的口味（recipe\_taste），食谱的烹饪方式（recipe\_craft），食谱的烹饪时间（recipe\_ntime），食谱的难度级别（recipe\_level）。

在后续的模型训练和实现中，会通过参考文献的处理数据的方式更正和完善本项目前期得到的数据，此处程序作为构建本项目知识图谱的模板来使用。

## （五）数据库设计

### 1、E-R 图设计

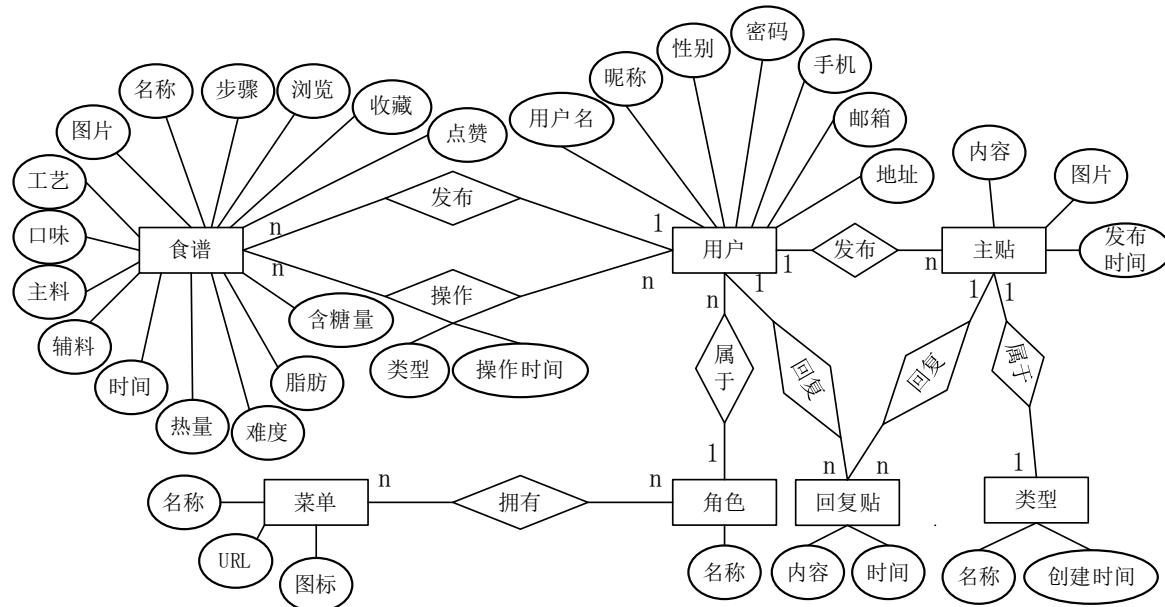


图 6 E-R 图

如图 6, 为本项目的 E-R 图设计, 其中包含食谱、用户、角色、菜单、主贴和回复贴六种实体, 有八种关系。

### (六) 后端项目搭建

后端项目使用 IDEA 为开发工具, 使用 SpringBoot 框架搭建, 运用 Mybatis 进行 MySQL 数据持久化操作, 使用 Redis 缓存减轻 MySQL 频繁查询的负担, 使用 Spring Security 作为认证和授权的工具, 保证接口的安全性和稳定性。本项目使用了 JSON Web Token (JWT) 来生成 token, 实现基于令牌的身份验证和授权, 即客户端在登陆成功后, 服务器返回一个 JWT 令牌, 客户端将该令牌存储在本地, 以后每次请求时将作为请求头部发送给服务器, 服务器根据该令牌验证用户身份和权限。

1、创建 SpringBoot 项目, 配置 MySQL 数据源, 配置 JWT 的请求头、加密密钥等参数, 配置 Spring Security 的方法, 如密码加密、认证过滤器、访问接口异常的自定义返回结果等, 配置 Swagger2。由于本项目是前后端分离的, 可以使用 Swagger2 来测试和查看 Controller 接口文档, 以便项目的开发。

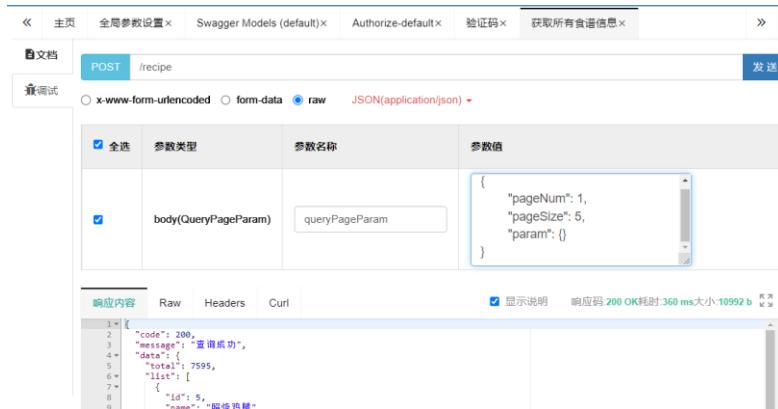


图 7 Swagger2 演示图

2、依据创建 MySQL 数据库表, 生成项目所使用的 pojo、controller、mapper、service 等包下的类。

如图 8 所示, 为后端项目目录结构图。从目录结构可以看出, 本项目是一个使用 SpringBoot 框架的 Maven 项目。其中 src.xz.server 包中为系统后台的代码。其中 pojo 包中存放了 Model 层的代码, 定义了与数据库中存储的数据对应的类。mapper 包中存放了操作 MySQL 数据库和 redis 的公开代码。service 包中的代码对应业务逻辑层, 实现系统中的各种业务逻辑。controller 包中的代码对应控制层, 接收前台请求, 并调用 service 中代码获取结果, 将结果反馈给前台。resources 主要存放系统的静态资

源、前端上传的文件和图谱、XML 配置文件等内容。

为了统一接口的返回数据类型，定义了公共返回对象，包含多个返回结果的静态成员函数，其成员变量如表 2 所示。

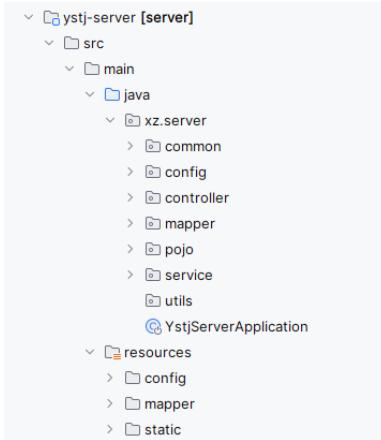


图 8 项目目录结构

表 2 RespBean 公共返回对象

变量名	描述
long code	返回代码。200: 成功 500: 失败
String message	返回描述。如“登录成功”、“注册成功”
Object data	返回数据
boolean state	是否将描述信息显示（前端使用 ElMessage 显示）

### （七）前端项目搭建

本项目前端主要使用 Node.js、Vue、Axios、router、Element-Plus 进行项目的搭建和编写。首先，配置 vite.config.js 解决跨域问题，跨域前缀为“/api”，当请求识别到跨域前缀时，则需要进行跨域。

项目需要配置响应拦截器，依据后端返回的 RespBean 公共返回对象的内容来执行响应拦截器的内容。其中，使用 ElMessage 来显示返回描述，给用户良好的体验。依据 axios 请求的构造形式，配置 post、put、get 和 delete 请求，请求参数为 url 和 params。最后，配置好路由模板，以便后续增添页面。

### （八）功能实现

#### 1、登录与注册

在登录注册页面中，首先后端生成验证码图片，并将结果存入 session 中，其次将验证码图片返回到页面中。需要注意的是，为了满足验证码的时效性，在前端获取

### 附录 3 中期报告

验证码图片时，需要在 URL 的末端加入时间来加以区别，以防止页面造成无法刷新获取新验证码的问题。在前端中，登录和注册页面的输入框都使用了正则表达式进行表单验证，如密码长度不少于 6 位，邮箱的格式等要求。



图 9 登录注册页面

在前端用户点击登录或注册时，会提交封装的 axios 请求到后端进行账号的登录或注册的业务逻辑操作。例如，登录操作的请求 url 为 “/api/login”，其中 “/api” 为跨域的前缀，当前端识别到跨域前缀时，进行跨域操作；“/login” 则为后端的 controller 层提供的 Restful 接口以供使用。

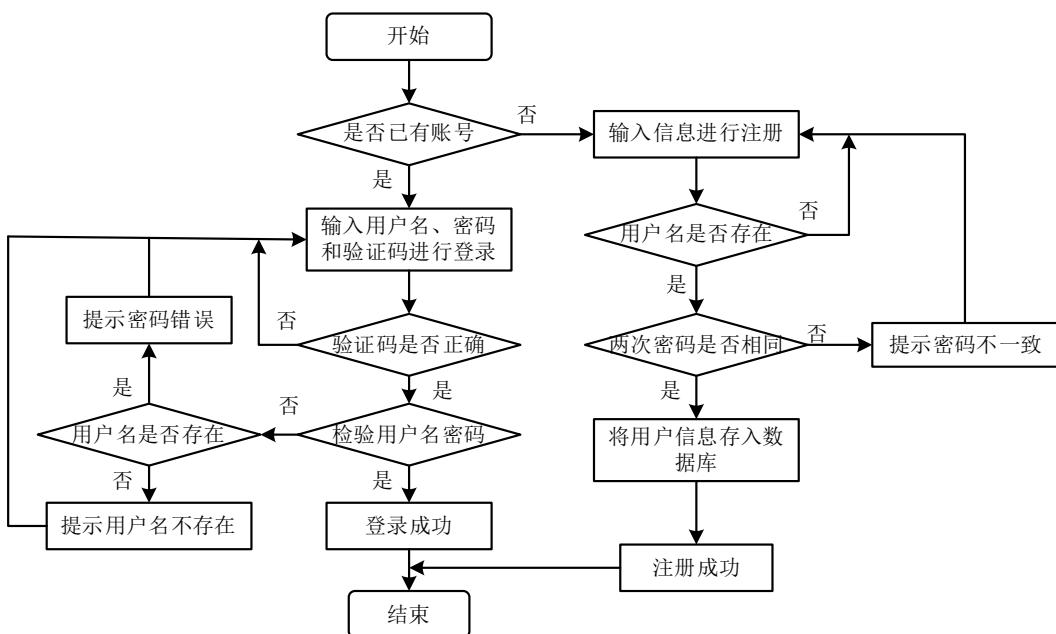


图 10 登录注册流程图

如图 10 所示，为系统的登录注册流程图。当用户已有账号时，则可以通过登录页面进行登录。前端将用户名、密码和验证码进行封装，通过请求发送给后端，后端则依次校验验证码、用户名和密码是否正确，如果不正确则向前端返回必要的提示以引导用户进一步的操作。当用户没有账号时，则可以通过注册页面进行注册，前端将一些通过表单验证的必要信息进行封装，发送注册请求给后端，后端则要先从数据库中判断用户名是否存在。如果存在，则将用户信息进行保存到数据库，注册成功；如果不存在，则返回必要的提示信息。

在用户信息方面，密码内容必须要做到加密，以提高系统的安全性和稳定性。本项目在获取到用户注册的信息时，会使用 Spring Security PasswordEncoder 进行密码加密，再存储到数据库中。当登录时，也是用其提供的接口进行匹配。在查询用户信息返回前端时，密码变量也设置为 NULL 再进行返回。

## 2、接口访问权限

后端需要限制未登录用户可以访问哪儿些接口，而登录用户可以访问哪儿些接口，Spring Security 通过配置权限忽略项可以配置开放的接口。对于某些特定场景，如食谱界面，未登录用户可以进行查看，当未登录用户执行点赞操作时，则提醒游客未登录，并跳转到登录页面。这一类的操作则需要使用过滤器来完成，希望这个过滤器禁止未登录访问需要登录的接口时，返回必要的提示信息。

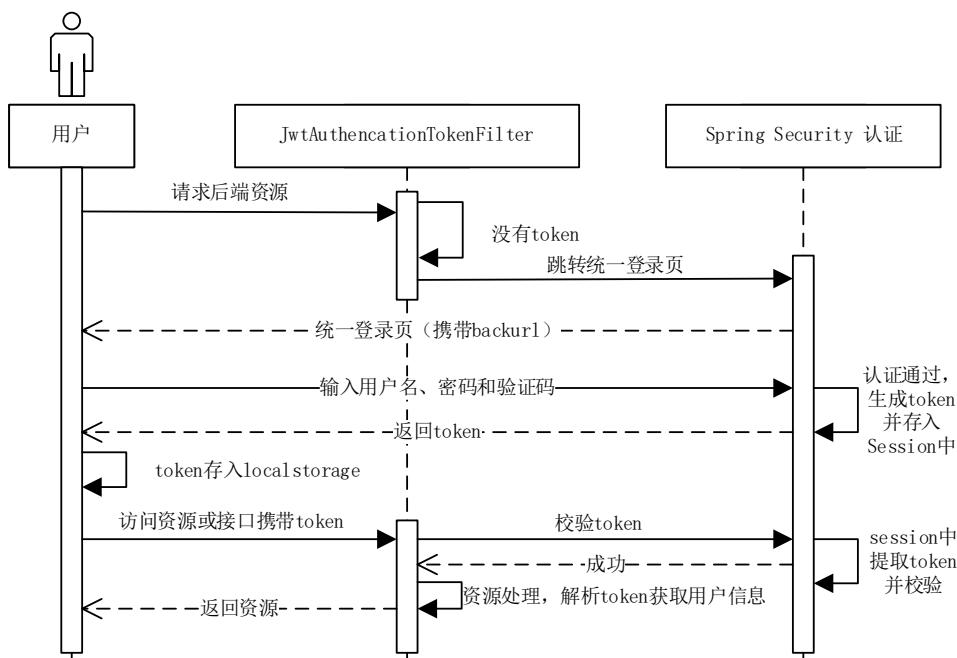


图 11 系统请求时序图

如图 11 所示，为系统请求时序图。当用户访问需要登录的后端资源时，过滤器会检测到请求中没有 token，则通过认证中心返回统一的登录页面的提示信息。用户则需要进行登录，当用户认证成功后，将生成的 token 存入 Session 中，并返回 token。前端会将 token 存入到 localstorage 中。前端的过滤器会再每次发送请求时，将 token 封装到请求中。当用户再次发送请求时，此时过滤器需要从 session 中提取 token 并进行校验。在校验成功后，进行资源的处理，并解析出用户信息，最终向前端返回请求的资源内容。

如上所述，为封装的请求的过程。未登录用户即可访问的接口是通过 Spring Security 的设置可以绕过 JwtAuthenticationTokenFilter 的认证。

### 3、查询分页

由于食谱和其他信息众多，免不了出现分页的情况，所以此处先封装好分页的实体类，为后续的查询分页操作提供基础。本项目主要采用 PageHelper 插件进行分页，PageHelper 的主要原理为在 Spring Boot 启动时，为我们的 SqlSessionFactory 添加了 PageInterceptor，这个拦截器会在 SQL 执行之前添加分页信息，查询完成后，可以将查询结果加上分页信息封装为 PageInfo 对象传给前端来解析。

表 3 QueryPageParam 分页实体类

属性	变量名	描述
Integer	pageSize	分页的大小（默认为 20）
Integer	pageNum	页号（默认为 1）
HashMap<String, Object>	param	接收前端传来的其它参数

如表 3 所示，为分页实体类，前端可以根据该类来进行参数的构造，传递页号、分页大小和其他参数即可获得分页的结果。前端可以根据返回的 PageInfo 对象来设置页号、分页大小和总记录数，从而达到分页的效果。

### 4、UserController

SpringBoot 项目有各个层，如 Controller、Service、Mapping 层。其中，Mapping 层主要负责对数据进行持久化操作，Service 层主要负责一些业务逻辑的处理，而 Controller 层负责对前端传输的数据进行处理。本项目主要采取的是厚 Controller 层的方式，业务逻辑处理都在 Controller 中进行处理，Service 层只负责一些增删改查的方法，这样可以避免 Service 层的嵌套造成错误。所以，本文也主要针对 Controller 所

出现的方法和业务逻辑流程进行介绍和说明。

在 UserController 中，主要实现了用户和管理员等不同角色对用户进行增删改查的操作，如表 4 为实现的各种接口和主要参数。其所有的返回对象皆为上述封装 RespBean 的实体类。

表 4 UserController 接口表

URL 地址	参数类型	描述
GET /user/{id}	String id	查询 id 为某值的用户信息
GET /user/getAll	QueryPageParam param	查询所有用户信息（分页）
GET /user/getUsersByLike	QueryPageParam param	模糊查询用户信息（分页）
GET /getUsersByRole	QueryPageParam param	根据角色查询用户（分页）
POST /user/save	String json	添加用户，参数有用户和角色信息
PUT /user/update	String json	更新用户，参数有用户和角色信息
PUT /user/updateRole	String json	修改用户角色，参数有用户 id 和角色 id
DELETE /user/delete/{id}	String id	删除 id 为某值的用户信息
PUT /user/updatepass	Map<String, String> map	修改密码，参数有新密码和旧密码
PUT /user/resetpass	Map<String, String> map	重置密码，参数有用户 id 和新密码

如表 4 所示，为 UserController 的接口表。其中包含用户可以查询、修改自己的个人信息和修改自己的密码，也可以查看其他用户的个人信息。管理员可以对用户信息进行模糊查询和精准查询，也可以重置用户的密码。

值得注意的是，修改密码的请求仅限于用户能够修改自己的密码，所以其中的请求参数并不包含用户的 id，用户的身份信息隐藏在 token 中，而管理员可以重置用户的密码的密码，参数需要用户的 id 和新密码。有关于新密码和旧密码是否重复等问题均在前端表单验证时实现。

## 5、LoginController

该层控制器主要负责提供当前登录用户一些逻辑的服务，其中涉及到的实体类有 UserLoginParam，其中包含用户名、密码和验证码三个变量。这个实体类仅用户用户登录使用，避免滥用 User 对象和其中的变量，造成冗余和程序错误等问题，简化前端登录请求参数的构造。

### 附录 3 中期报告

在此控制器中，主要包含登录返回 token、注册用户信息和获取当前用户信息的功能，在以后有关当前用户信息的有关接口将会在此处定义。

表 5 LoginController 接口表

URL 地址	参数类型	描述
POST /login	UserLoginParam p	登录，如果登录成功，则返回 token
POST /REGISTER	User user	注册用户信息，如果注册成功，则返回 token
GET /getuserinfo	Principal p	返回当前登录用户信息

如表 5 中，在返回当前登录用户信息的接口参数是 Principal 类型，它是 Spring Security 通过验证后提供的变量，无需前端进行构造返回。通过 principal 即可获取登录时，保存在其中的用户名，进而得到当前登录的用户信息。

### 6、RecipeController

该控制器主要处理有关食谱增删改查操作，例如获取所有食谱信息，获取某个食谱信息，进行模糊查询，删除食谱信息，批量删除等操作。如表 6 所示，为目前已经实现的 RecipeController 接口。

表 6 RecipeController 接口表

URL 地址	参数类型	描述
GET /recipe	QueryPageParam param	查询所有食谱信息（分页）
GET /recipe/like	QueryPageParam param	模糊查询，名称、工艺、口味、食材
GET /recipe/{id}	String id	查询 id 为某值的食谱信息
DELETE /recipe/{id}	String id	删除 id 为某值的食谱信息
DELETE /recipe/deleteIds	Integer[] ids	删除 ids 中的食谱信息
POST /recipe/add	Recipe recipe	添加食谱信息
GET /getAllCategories	—	获取所有的类别 (工艺、口味、所需时间，难度)

### 7、Redis 与 MySQL

点赞和收藏是本项目的基本功能，其中在食谱信息页码可能涉及到显示点赞数量，判断用户是否点赞，显示个人点赞列表等内容。例如，当点赞某食谱的用户再次进入到某页面中，需要显示该用户已经点赞了该食谱，由于食谱点赞数据表数据众多，如果频繁查询则可能会造成网站速度加载缓慢的情况。

如图 12 所示，为读操作的场景中处理流程图。当发生读操作时，先从 Redis 中查看是否有该类型的 key，如果没有则从 MySQL 中加载数据后更新到 Redis 中，如果有则返回所得到的数据。

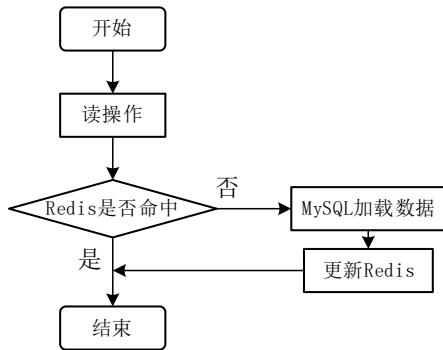


图 12 读操作处理流程图

如图 13 所示，为写操作的场景中处理流程图。当发生写操作时，查看 Redis 是否存在该类型的 key，如果没有则更新 Redis 中，如果有，则将其取出进行修改后再放入 Redis。随后，在项目的定时任务中，再将缓存中的数据写入 MySQL 中。

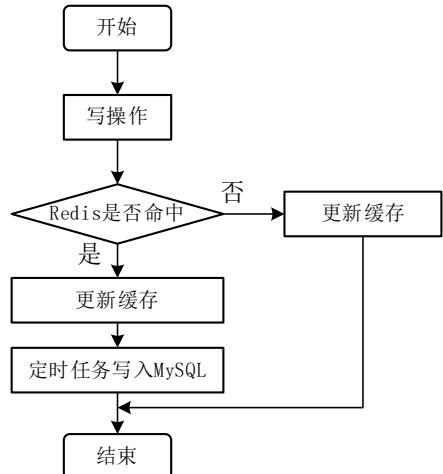


图 13 写操作处理流程图

在上述的 RecipeController 中请求 “/getAllCategories” ，由于获取的类别是从食谱表中进行去重查询出来的，分别要查询工艺、口味、所需时间和难度四个字段的去重值，查询耗时长。引入 Redis 后，服务器可以在开始时，启动任务优先将获取类别查询出来存入 Redis 中，后续再进行查询时直接从 Redis 中获取即可。

## 8、RecipeOpController

RecipeOpController 控制层主要实现对食谱进行点赞和收藏的操作。在点赞和收藏的操作中，都涉及到以上的 Redis 读和写的处理流程。例如，当用户获取某个食谱

### 附录 3 中期报告

信息后，先查询 Redis 是否有该食谱信息的点在列表，如果有，则判断该食谱的点赞列表是否存在该用户，如果存在，则返回该用户点赞了该食谱；如果不存在，则加载该食谱的所有点赞用户 id 到 Redis 中。如表 7 为该控制层已经实现的接口。

表 7 RecipeOpController 接口表

URL 地址	参数类型	描述
PUT /recipeop/star/{id}	String id, Principal p	当前登录用户点赞 id 为某值的食谱
PUT /recipeop/nostar/{id}	String id, Principal p	当前登录用户取消点赞 id 为某值的食谱
PUT /recipeop/collect/{id}	String id, Principal p	当前登录用户收藏 id 为某值的食谱
PUT /recipeop/nocollect/{id}	String id, Principal p	当前登录用户取消收藏 id 为某值的食谱

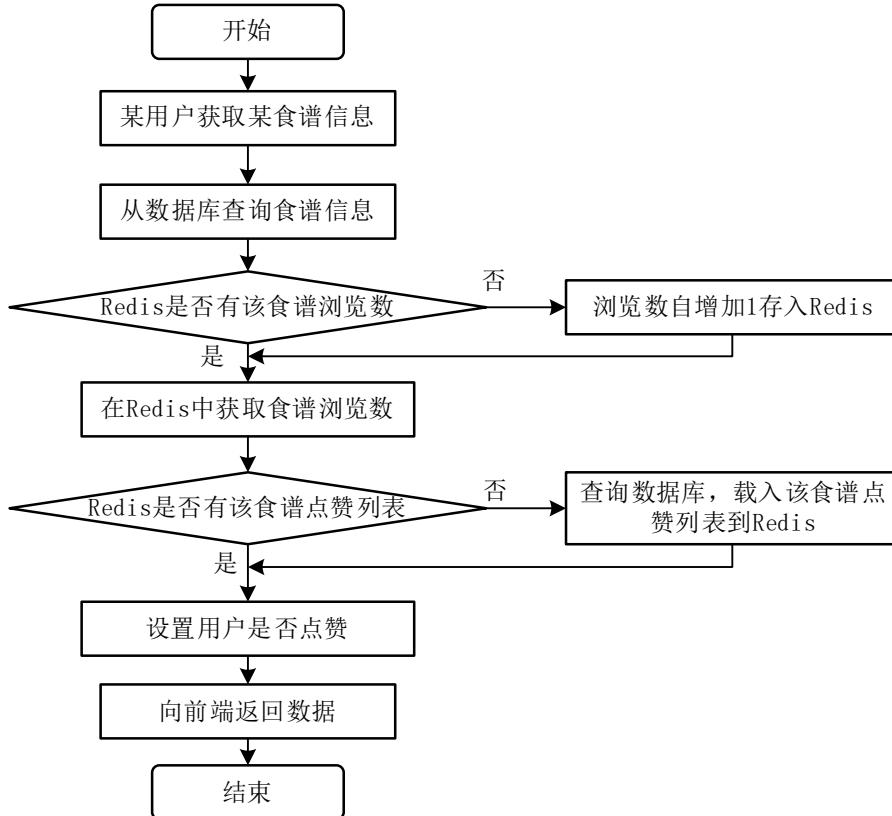


图 14 用户获取食谱流程图

如图 14 所示，为用户获取食谱信息的流程图，其中设计点点赞列表，未包含收藏列表，收藏列表和点赞列表同理，没有在图上绘制。

每一个接口均依照读、写处理流程进行编写代码。由于本项目后续要获取到精准的点赞和收藏数据，所以要保证点赞去重，避免重复点赞，要实现这个需求，必须要

有文章点赞的 uid 列表，同理，收藏也是这样。如表 8 所示，为本项目中设计的存储在 Redis 的 key 的构造表。

表 8 Redis key 构造表

Redis key 值	value 类型	value 描述
recipe:browse:num:rid:{recipeId}	string	recipeId 的食谱的浏览量
recipe:star:num:rid:{recipeId}	string	recipeId 的食谱点赞量
recipe:star:uidList:rid:{recipeId}	list	recipeId 的食谱点赞的用户列表
recipe:collect:num:rid:{recipeId}	string	recipeId 的食谱收藏量
recipe:collect:uidList:rid:{recipeId}	list	recipeId 的食谱收藏的用户列表

表 8 中的 recipeId 为食谱的 id。在进行查询时，如果没有某个食谱的 Redis key 值，则可以将该食谱的相关点赞和收藏内容从 MySQL 数据库中进行查询，并将其存入到 Redis 中。

如图 15 所示，为用户点赞食谱时序图。当用户点赞食谱时，过滤器会首先解析用户发送请求所携带的 token，获取到用户的信息，其次进入控制层。控制层调用服务层的函数，首先判断 Redis 是否含有构造的 key，如果有则进行加 1 操作，并将用户 id 插入到该食谱的点赞列表中；如果没有则从数据库中查询该食谱的点赞情况和点赞列表，并进行加 1 的操作，最终返回点赞量。同理，用户收藏食谱也是使用该方式实现的。

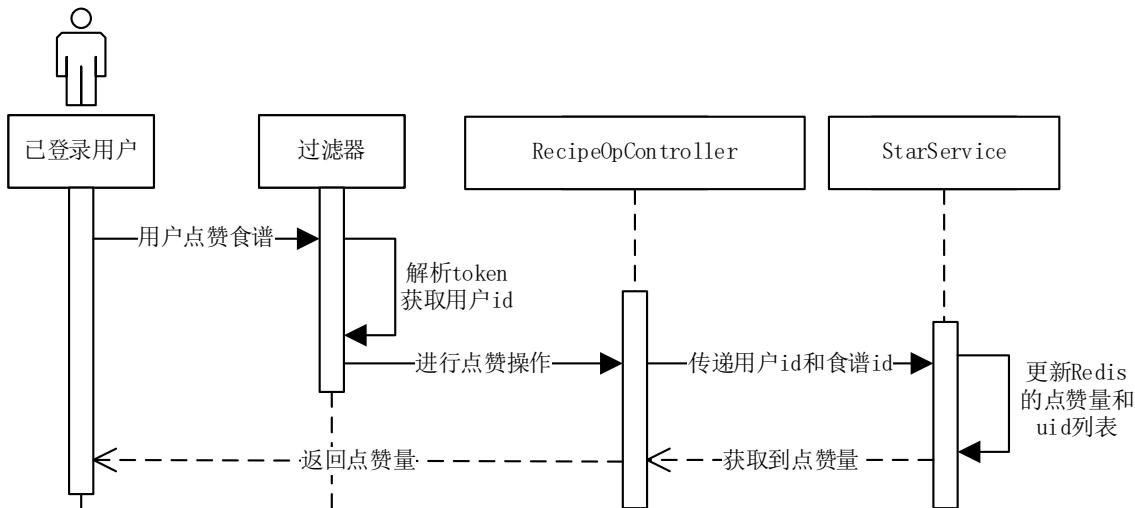


图 15 用户点赞食谱时序图

在用户获取食谱信息时，浏览量、该用户是否收藏和点赞，均会被从 Redis 中获取到，返回到前端，从而进行显示。

## (九) 模型推荐

本文主要使用一种 RippleNet 模型。该模型主要使用了偏好属性多样性指标，对 RippleNet 模型的知识图谱的采样方法进行了改进。将用户的食谱喜好在知识图谱上的扩展的过程中的固定邻居个数的随机实体采样方法改成了基于知识图谱扩展偏好集合共有属性的随机采样，提高了采样集包含的用户兴趣物品属性信息量。

### 1、基于 RippleNet 模型的饮食推荐

#### (1) 模型框架

RippleNet 模型主要分为采样训练模块和预测模块两部分。

采样训练模块使用共有属性随机采样替代 RippleNet 模型扩展过程中的固定邻居个数的随机采样。通过共有属性随机采样重构偏好特征，使得内部节点间的相关性得到增强，用户的偏好特征包含的信息量得到提高。

预测模块通过加载上述模块训练得到的模型参数，通过重构用户  $U$  历史收藏的食谱  $V$  的特征表示，计算用户点击其他食谱的概率，并根据概率预测出用户潜在的兴趣。

#### (2) Hop-n 构造

用户  $u$  的第  $k$  跳的波纹集合被定义为头节点  $h$  在  $\varepsilon_u^{k-1}$  中的尾结点的集合：

$$\varepsilon_u^k = \{t \mid (h, r, t) \in \mathcal{G} \text{ and } h \in \varepsilon_u^{k-1}\}, \quad k = 1, 2, \dots, H \quad (1)$$

如公式(1)所示，特别的当  $k=0$  时， $\varepsilon_u^0$  表示用户当前收藏的食谱序列。当  $k=1$  时，表示以用户收藏的序列为头实体的三元组的尾实体。其他情况依次类推，可以得到  $\varepsilon_u^k$  就是第  $k$  个波纹上的实体。如上述可得，用户  $u$  的第  $k$  跳的波纹集合的三元组为：

$$S_u^k = \{(h, r, k) \mid (h, r, t) \in \varepsilon_u^{k-1}\}, \quad k = 1, 2, \dots, H \quad (2)$$

RippleNet 的用户特征聚合了多条的节点特征和路径特征，并且能够给出相关性最高的路径作为可解释的推荐结果。 $h$  和  $v_i$  在关系空间  $R_i$  的相似度也叫做相关概率  $p_i$ ，需要使用的时 1-hop (知识图谱中第一次向外扩展的食谱信息) 的 RippleSet，对于 set 中的每一个  $(h, r, t)$ ，用  $(h * r)$  与 item  $v$  相乘得到 item  $v$  与 1-hop 里每个  $(h_i, r_i)$  的相关性得分，再通过 softmax 进行相关性得分的归一化，计算公式如下所示：

$$p_i = \text{softmax}(v^T R_i h_i) = \frac{\exp(v^T R_i h_i)}{\sum_{(h, r, t \in S_u^1)} \exp(v^T R_i h_i)} \quad (3)$$

用户兴趣特征  $u$  的计算公式如下所示：

$$u = \sum p_i t_i \quad (4)$$

最后，结合用户特征和物品特征计算预测的点击概率：

$$y_{uv} = \text{sigmoid}(u^T v) \quad (5)$$

由于本项目的知识图谱中食谱指向主料、辅料等属性，而主料并非指向食谱，所以需要规定偏好传播过程中的路径实体类型。用户兴趣在构建的食谱知识图谱传播过程中，奇数跳尾节点代表用户感兴趣食谱的属性实体，偶数跳尾结点代表偏好属性对应的食谱实体。

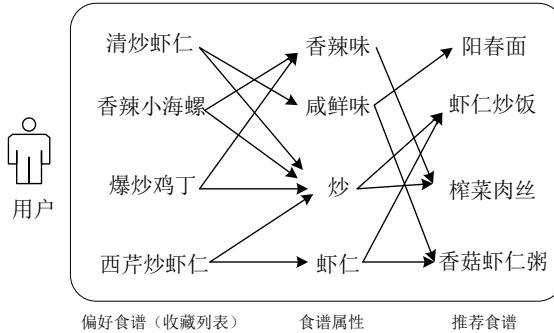


图 16 用户感兴趣食谱共有属性

如图 16 所示，一跳的波纹集合用户的偏好的属性实体为：香辣味、咸鲜味、炒和虾仁等。二跳波纹集合用户的扩展偏好食谱实体为：阳春面、虾仁炒饭、榨菜肉丝和香菇虾仁粥。每个食谱属性偏好在提取出的三元组中的(r,t)组合特征共有频次代表了用户对该偏好食谱的偏爱程度。

用户兴趣经过两跳以上的传播，由于相关概率  $p_i$  小于 1，三跳及以后的偏好与种子相关性急剧下降且 RippleSet 的数量呈爆炸级别增长，增加了额外工作量。所以，此项目主要选择跳数  $H = 2$ 。

在根据本项目构建知识图谱的特点，依据 RippleNet 的结构和源码，进行编写模型训练的程序。由于获取数据没有评分、不喜欢的数据，所以使用随机取样的方式获取负样本数据，进而训练模型。在经过实验发现，训练集的准确率趋于稳定在 90% 以上，而测试集的准确率在 65% 左右，其对于该项目来说，基本跟随机选择没什么区别，效果甚微。

经查阅资料，推测其可能出现的问题是样本数据缺乏准确的结果，而采用随机取样的方式，不适合本项目的数据集。另一个主要原因可能是由于本项目知识图谱的数据集形式过于单一，且数据中存在较多的干扰数据，导致记录了很多噪声特征，可能存在数据不准确的现象。由此需要进行改善采样方式。

#### (4) 存在待问题和改进方案

通过查询食谱交互数据集得知，交互记录数为 55367，用户数为 1926，食谱数为 5708，平均每个用户有 20 条交互记录。根据数据稀松度计算公式计算得到食谱数据集稀松度为 0.3%，即无评分数据占据整个样本空间 99.7%，要从 0.3% 的用户-食谱交互记录数据学习用户偏好，数据集太过稀疏，模型不太容易学习到用户偏好特征，容易欠拟合。数据集的数据稀疏主要是由于用户和食谱的交互不充分，平均每个食谱的收藏数和平均每个用户收藏过的食谱数远远小于用户数和食谱数。食谱数据集稀疏度的计算如公式(1)所示：

$$\text{数据集稀疏度} = \frac{\text{总交互记录}}{\text{用户数} \times \text{食谱数据}} \quad (6)$$

用户兴趣在知识图谱的传播过程中存在像水波纹扩散一样的现象。随着波纹扩散，波纹逐渐衰减，但存在多个波源时，用户的偏好物品扩散出的波纹集会产生多个波纹，波纹间通过偏好物品间的共有属性产生联系相互干涉叠加形成干涉加强区域，即下图中深色节点集合。

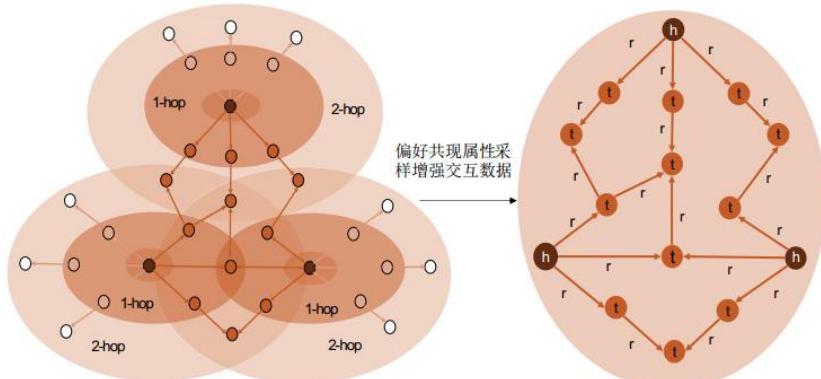


图 17 用户感兴趣食谱共有属性

如图 17 所示，通过在知识图谱地多跳扩展上地食谱共现属性采样，可以获得用户的偏好扩展，将之视为用户收藏的食谱对用户偏好数据集进行增强。用户偏好数据增强算法如表 9 所示。

表 9 用户偏好数据增强算法

#### 算法 1. 用户偏好数据增强算法

**输入：**交互记录  $\mathbf{Y}$ 、知识图谱  $\mathbf{G}$ 、参数  $k$

**输出：**交互记录  $\mathbf{Y}$

1. 在知识图谱上为每个用户随机采样波纹集合  $\{s_u^k\}_{k=2}^H$

- 
2. **for**( $h, r, t$ ) in  $\{s_u^k\}_{k=2}^H$  **do**
  3.     统计不同  $h$  共有的( $r, t$ )类别数  $m$  和频数  $n_{uc}$
  4.     **If**  $n > k$  **and**  $t$  不在交互记录  $Y$  的正样本中 **then**
  5.         将  $t$  加入交互记录  $Y$  正样本中
  6.     **Else if**  $n = 0$  **and**  $t$  不在交互记录  $Y$  的负样本中 **then**
  7.         将  $t$  加入交互记录  $Y$  负样本中
  8. **end for**
  9. **return**  $Y$
- 

以上是本项目基于 RippleNet 模型的预测和搭建的工作进展，测试集准确率在 65% 左右，准确率较低。后续，会根据上述所提到的用户偏好数据增强算法来改善样本选择的方案，处理原始数据噪声较大的数据内容。

## 2、基于 Neo4j 个性化饮食推荐

本项目知识图谱通过 Python 程序、Neo4j 工具半自动构建，使用 Excel 表格的数据构建知识图谱。Neo4j 使用图模型建模数据，内置了许多图算法，在可视化食谱知识图谱的前提下，可以快速挖掘知识图谱在图结构上的潜在关联。

### 基于入度排序的饮食推荐

本项目在使用 Neo4j 构建的知识图谱中，食谱节点的出度表示食谱节点指向属性节点的关系边的数量。属性节点的入度代表不同食谱节点指向同一属性节点的关系边的数量，从一定程度上可以描述属性节点在不同的食谱之间的同现频率，代表该节点在知识图谱中的热度，反应了整体用户群体的属性偏好分布。

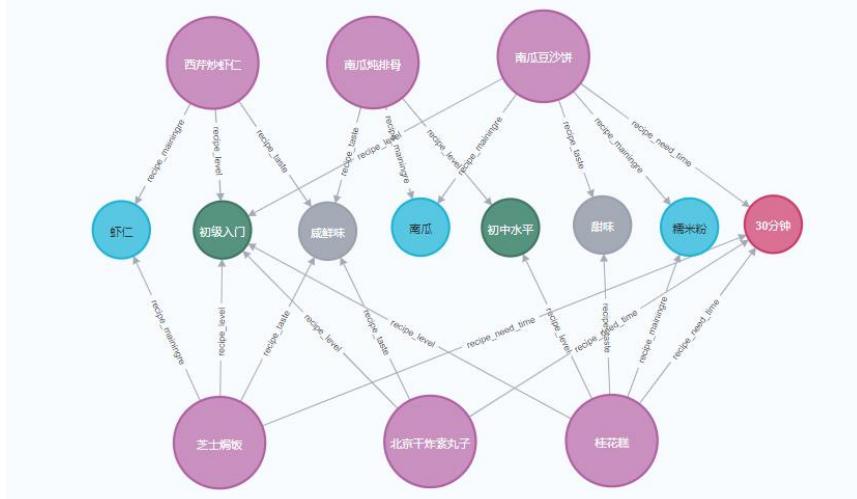


图 18 基于入度排名进行食谱推荐

### 附录 3 中期报告

如图 18 所示，选取用户 A 收藏的 6 个食谱作为案例进行分析。节点“咸鲜味”的入度为 4，节点“初级入门”的入度为 5，节点“虾仁”、“南瓜”、“糯米粉”的入度均为 2，节点“30 分钟”的入度为 4。通过分析用户收藏的食谱所设计的知识图谱各个节点的入度，可以得出用户上传的食谱共有属性的排名。综上可得，该用户的偏好口味是咸鲜味和甜味，并且喜欢咸鲜味胜过甜味，所以在推荐时推荐这两种口味的食谱给用户并且有限推荐食谱属性中包含入度数大的属性的食谱。

表 10 所有用户食谱口味热度表（8 项）

node_id	node_labels	node_name	in_degree
11822	taste	家常味	2254
11841	taste	咸鲜味	1524
11823	taste	香辣味	692
11835	taste	甜味	502
11821	taste	酸甜味	250
11838	taste	酸辣味	151
11829	taste	酱香味	136
11842	taste	黑椒味	44

从表 10 所示可得，美食街网站的用户上传的食谱，基本反应了中国人的饮食偏好。如排在前列的家常味、咸鲜味、香辣味等口味是典型的中式菜系的口味。鸡蛋、面粉、猪肉、五花肉、胡萝卜、尖椒、土豆等是食谱中较为常见的食材。

表 11 查询知识图谱入度排名前 10 的主食材

#### Cypher 查询语句

```
MATCH (p:recipe)-[r]->(t:main_ingredient)
WITH t, count(t) as inDegree
ORDER by inDegree DESC
RETURN t.name, inDegree
LIMIT 10
```

如表 11 所示，为查询知识图谱入度排名前 10 的主食材的 Cypher 查询语句。基于入度排名的方法提供了一种解决冷启动的推荐方法。对于新注册到系统的用户，可以通过分析热度较高的食谱推荐给用户。不同于一般的流行推荐，通过 Neo4j 的入

度出度查询排名，不仅可以获得较为热门的食谱，还可以获得最热门的食材、食谱口味等，能够从多角度出发展示热门食谱。对于已注册一段时间的用户，可以在总体食谱知识图谱上匹配出用户浏览记录的子图，然后在该用户浏览记录的子图使用初读排序，分析该用户的个性化饮食偏好属性。

### （2）基于相似度的食谱推荐

使用 Neo4j 数据库内置的 NPOC 插件计算相似度可以很方便的挖掘食谱节点与属性节点之间的潜在关系。

表 12 查询与可乐鸡翅食材相似度排名前 10 的食谱

---

#### Cypher 查询语句

```
WITH "可乐鸡翅" AS targetRecipeName
MATCH(p1:recipe {name:targetRecipeName})-[r1]->(t1:main_ingredient)<-[r2]-(p2:recipe)-
[r3]->(t2:main_ingredient)
WHERE r1<>r2
WITH p2,
    apoc.text.levenshteinDistance(t1.name, t2.name) AS distance
ORDER BY distance ASC
RETURN p2.name AS recommRecipe, distance LIMIT 10
```

---

如表 12 所示为查询语句，通过匹配从食谱“可乐鸡翅”节点出发经过食谱属性节点“t1”，经过第二个食谱节点“p2”，最后指向第二个食谱属性节点“t2”的所有路径，使用文本相似度函数计算食谱属性的“t1”和“t2”的相似度为排序依据进行排序，将用户的交互记录中的偏好在知识图谱上食谱主食材的属性上进行了两跳的传播，以挖掘用户潜在偏好的食谱。

## 三、存在的问题和拟解决方法

### 1、数据获取

数据获取时，速度较慢，偶尔可能会被限制访问。后续在爬取数据时，降低爬取频次和速度，稳步安全获取数据。获取的数据可能会出现很多空值、无意义的值、错误信息等，这样的情况可以多整理错误信息，使用正则表达式来进行数据的筛选和剔除，以获取更加准确完善的数据。

### 2、推荐模型的实现

通过阅读已有论文和其他开源代码，结合本项目知识图谱和食谱的字段设置，不断加强和完善食谱数据，继续进行推荐模型的训练和使用。现阶段已经通过 Flask 搭建好的服务器接口，只需等到推荐模型训练完成，即可进行调用使用。

### 3、前端页面

后端服务器已经根据现有的数据表，完成了 Controller、service 和 mapping 层的增删改查的操作，并且可以根据业务需求，设计实现了接口。这些实现接口均以通过 swagger2 模拟发送请求进行了验证。现阶段较大的难点是前端的数据展示和页面设计，在下一阶段的实施过程中，我会访问相关类型的网站，学习其他网站的页面设计思路和想法，并将其运用到本项目中。

### 4、图片的上传

在进行食谱信息的上传时，需要用到上传图片到后端服务器中，并将其保存到特定的路径上。而前端访问服务器路径上的静态资源需要 SpringBoot 进行相关的配置，现在该问题还不能有效解决，接下来会通过查阅相关的学习资料进一步学习和实现。

### 5、前端项目文件夹创建较乱

由于前期项目开发过程中，项目规划不到位，有个别文件夹和文件出现冗余，可以进行复用的组件没有进行复用而是重新编写代码，大大降低了开发效率。在下一阶段，会着重项目的规划，尽可能多地复用组件，为外界暴露接口，降低耦合。

### 6、项目进度

抓紧时间完成接下来的项目编码，尽可能多地撰写文档，为后期撰写毕业论文提供参考。

### 7、动态路由

由于管理员会有多个身份，每个身份都有相应可以操作的菜单选项，这个菜单可以通过后台最高管理员进行设置。目前可以实现根据身份能够赋予前端路由不同的值，现在需要完成前端后台管理系统页面设计完成之后，再进行测试。

### 8、编码逻辑易混乱

由于编码内容较多，有一些编码逻辑和业务流程容易忘记，可能会造成开发效率低的问题。在下一阶段，我会在编码过程中尽可能多得绘制时序图和流程图，以保证在以后的编码过程中头脑清晰，逻辑清楚。

## 9、毕业论文的撰写

前期忙于项目代码的编写，目前毕业论文的撰写工作开展较慢，在下一阶段的工作过程中，每天抽出 2~3 个小时的时间，开始毕业论文的准备和撰写工作。在毕业论文撰写的过程中，可能会出现对前期已完成工作有进一步的认识和看法，在不影响项目进度的情况下，对前期工作进行细微修改。

## 10、各个图的绘制和使用

自毕业设计开题以来，报告中所出现的一些图，因为绘制不规范可能需要第二次修改绘制，甚至是第三次修改绘制，而且有些部分缺乏流程图或其他图，文字解释显得苍白无力。所以，在下一阶段，如果遇到需要绘制图的部分，首先要调研类似图的标准样式再进行绘制，一次成功，以免出现二次绘制的现象。同时，能多用图来解释的部分，尽量用图来解释，这样可以更能通俗易懂，而且对自己理清整个编程逻辑有很大帮助。

## 附录 4 外文原文

To address the sparsity and cold start problem of collaborative filtering, researchers usually make use of side information, such as social networks or item attributes, to improve recommendation performance. This paper considers the knowledge graph as the source of side information. To address the limitations of existing embeddingbased and path-based methods for knowledge-graph-aware recommendation, we propose *RippleNet*, an end-to-end framework that naturally incorporates the knowledge graph into recommender systems. Similar to actual ripples propagating on the water, RippleNet stimulates the propagation of user preferences over the set of knowledge entities by automatically and iteratively extending a user's potential interests along links in the knowledge graph. The multiple "ripples" activated by a user's historically clicked items are thus superposed to form the preference distribution of the user with respect to a candidate item, which could be used for predicting the final clicking probability. Through extensive experiments on real-world datasets, we demonstrate that RippleNet achieves substantial gains in a variety of scenarios, including movie, book and news recommendation, over several state-of-the-art baselines.

The explosive growth of online content and services has provided overwhelming choices for users, such as news, movies, music, restaurants, and books. Recommender systems (RS) intend to address the information explosion by finding a small set of items for users to meet their personalized interests. Among recommendation strategies, *collaborative filtering* (CF), which considers users' historical interactions and makes recommendations based on their potential common preferences, has achieved great success. However, CF-based methods usually suffer from the sparsity of user-item interactions and the cold start problem. To address these limitations, researchers have proposed incorporating *side information* into CF, such as social networks, user/item attributes, images and contexts.

Among various types of side information, *knowledge graph* (KG) usually contains much more fruitful facts and connections about items. A KG is a type of directed heterogeneous graph in which nodes correspond to *entities* and edges correspond to

*relations*. Recently, researchers have proposed several academic KGs, such as NELL1 , DBpedia2 , and commercial KGs, such as Google Knowledge Graph3 and Microsoft Satori4 . These knowledge graphs are successfully applied in many applications such as KG completion, question answering, word embedding , and text classification.

Inspired by the success of applying KG in a wide variety of tasks, researchers also tried to utilize KG to improve the performance of recommender systems. As shown in Figure 1, KG can benefit the recommendation from three aspects: (1) KG introduces semantic relatedness among items, which can help find their latent connections and improve the *precision* of recommended items; (2) KG consists of relations with various types, which is helpful for extending a user’s interests reasonably and increasing the *diversity* of recommended items; (3) KG connects a user’s historical records and the recommended ones, thereby bringing *explainability* to recommender systems. In general, existing KG-aware recommendation can be classified into two categories.

The first category is *embedding-based methods*, which pre-process a KG with *knowledge graph embedding* (KGE) algorithms and incorporates the learned entity embeddings into a recommendation framework. For example, Deep Knowledge-aware Network (D KN) treats entity embeddings and word embeddings as different channels, then designs a CNN framework to combine them together for news recommendation. Collaborative Knowledge base Embedding (CKE) combines a CF module with knowledge embedding, text embedding, and image embedding of items in a unified Bayesian framework. Signed Heterogeneous Information Network Embedding (SHINE) designs deep autoencoders to embed sentiment networks, social networks and profile (knowledge) networks for celebrity recommendations. Embedding-based methods show high flexibility in utilizing KG to assist recommender systems, but the adopted KGE algorithms in these methods are usually more suitable for in-graph applications such as link prediction than for recommendation , thus the learned entity embeddings are less intuitive and effective to characterize inter-item relations.

The second category is *path-based methods*, which explore the various patterns of connections among items in KG to provide additional guidance for recommendations. For example, Personalized Entity Recommendation (PER) and Meta-Graph Based

Recommendation treat KG as a heterogeneous information network (HIN), and extract meta-path/meta-graph based latent features to represent the connectivity between users and items along different types of relation paths/graphs. Path-based methods make use of KG in a more natural and intuitive way, but they rely heavily on manually designed meta-paths, which is hard to optimize in practice. Another concern is that it is impossible to design hand-crafted meta-paths in certain scenarios (e.g., news recommendation) where entities and relations are not within one domain.

To address the limitations of existing methods, we propose *RippleNet*, an end-to-end framework for knowledge-graph-aware recommendation. RippleNet is designed for click-through rate (CTR) prediction, which takes a user-item pair as input and outputs the probability of the user engaging (e.g., clicking, browsing) the item. The key idea behind RippleNet is *preference propagation*: For each user, RippleNet treats his historical interests as a seed set in the KG, then extends the user's interests iteratively along KG links to discover his hierarchical potential interests with respect to a candidate item. We analogize preference propagation with actual ripples created by raindrops propagating on the water, in which multiple "ripples" superpose to form a resultant preference distribution of the user over the knowledge graph. The major difference between RippleNet and existing literature is that RippleNet combines the advantages of the above mentioned two types of methods: (1) RippleNet incorporates the KGE methods into recommendation naturally by preference propagation; (2) RippleNet can automatically discover possible paths from an item in a user's history to a candidate item, without any sort of hand-crafted design.

The knowledge-graph-aware recommendation problem is formulated as follows. In a typical recommender system, let  $U = \{u_1, u_2, \dots\}$  and  $V = \{v_1, v_2, \dots\}$  denote the sets of users and items, respectively. The user-item interaction matrix  $\mathbf{Y} = \{y_{uv} | u \in U, v \in V\}$  is defined according to users' implicit feedback, where  $y_{uv} = 1$ , if interaction  $(u, v)$  is observed; and  $y_{uv} = 0$ , otherwise.

A value of 1 for  $y_{uv}$  indicates there is an implicit interaction between user  $u$  and item  $v$ , such as behaviors of clicking, watching, browsing, etc. In addition to the interaction matrix  $\mathbf{Y}$ , we also have a knowledge graph  $G$  available, which consists of massive entity-relation-entity triples  $(h, r, t)$ . Here  $h \in E$ ,  $r \in R$ , and  $t \in E$  denote the head,

relation, and tail of a knowledge triple, respectively,  $E$  and  $R$  denote the set of entities and relations in the KG. For example, the triple (*Jurassic Park*, *film.film.director*, *Steven Spielberg*) states the fact that Steven Spielberg is the director of the film "Jurassic Park". In many recommendation scenarios, an item  $v \in V$  may associate with one or more entities in  $G$ . For example, the movie "Jurassic Park" is linked with its namesake in KG, while news with title "France's Baby Panda Makes Public Debut" is linked with entities "France" and "panda".

Given interaction matrix  $\mathbf{Y}$  as well as knowledge graph  $G$ , we aim to predict whether user  $u$  has potential interest in item  $v$  with which he has had no interaction before. Our goal is to learn a prediction function  $y^{\hat{u}v} = F(u, v; \Theta)$ , where  $y^{\hat{u}v}$  denotes the probability that user  $u$  will click item  $v$ , and  $\Theta$  denotes the model parameters of function  $F$ .

In this section, we evaluate RippleNet on three real-world scenarios: movie, book, and news recommendations [5]. We first introduce the datasets, baselines, and experiment setup, then present the experiment results. We will also give a case study of visualization and discuss the choice of hyper-parameters in this section.

Since MovieLens-1M and Book-Crossing are explicit feedback data, we transform them into implicit feedback where each entry is marked with 1 indicating that the user has rated the item (the threshold of rating is 4 for MovieLens-1M, while no threshold is set for Book-Crossing due to its sparsity), and sample an unwatched set marked as 0 for each user, which is of equal size with the rated ones. For MovieLens-1M and Book-Crossing, we use the ID embeddings of users and items as raw input, while for Bing-News, we concatenate the ID embedding of a piece of news and the averaged word embedding of its title as raw input for the item, since news titles are typically much longer than names of movies or books, hence providing more useful information for recommendation.

We use Microsoft Satori to construct the knowledge graph for each dataset. For MovieLens-1M and Book-Crossing, we first select a subset of triples from the whole KG whose relation name contains "movie" or "book" and the confidence level is greater than 0.9. Given the sub-KG, we collect IDs of all valid movies/books by matching their names with tail of triples (*head, film.film.name, tail*) or (*head, book.book.title, tail*). For simplicity, items with no matched or multiple matched entities are excluded. We then match the IDs

with the head and tail of all KG triples, select all well-matched triples from the sub-KG, and extend the set of entities iteratively up to four hops. The constructing process is similar for Bing-News except that: (1) we use entity linking tools to extract entities in news titles; (2) we do not impose restrictions on the names of relations since the entities in news titles are not within one particular domain.

We compare the proposed RippleNet with the following state-of-the-art baselines:

- CKE [43] combines CF with structural knowledge, textual knowledge, and visual knowledge in a unified framework for recommendation. We implement CKE as CF plus structural knowledge module in this paper.
- SHINE [32] designs deep autoencoders to embed a sentiment network, social network, and profile (knowledge) network for celebrity recommendation. Here we use autoen coders for user-item interaction and item profile to predict click probability.
- DKN [33] treats entity embedding and word embedding as multiple channels and combines them together in CNN for CTR prediction. In this paper, we use movie/book names and news titles as textual input for DKN.
- PER [42] treats the KG as HIN and extracts meta-path based features to represent the connectivity between users and items. In this paper, we use all item-attribute-item features for PER (e.g., “movie-director-movie”).
- LibFM [20] is a widely used feature-based factorization model in CTR scenarios. We concatenate user ID, item ID, and the corresponding averaged entity embeddings learned from TransR [14] as input for LibFM.
- Wide&Deep [6] is a general deep model for recommendation combining a (wide) linear channel with a (deep) nonlinear channel. Similar to LibFM, we use the embeddings of users, items, and entities to feed Wide&Deep.

In RippleNet, we set the hop number  $H = 2$  for MovieLens-1M/BookCrossing and  $H = 3$  for Bing-News. A larger number of hops hardly improves performance but does incur heavier computational overhead according to experiment results. The complete

hyper-parameter settings are given in Table 2, where  $d$  denotes the dimension of embedding for items and the knowledge graph, and  $\eta$  denotes the learning rate. The hyper-parameters are determined by optimizing  $AUC$  on a validation set. For fair consideration,

the latent dimensions of all compared baselines are set the same as in Table 2, while other hyper-parameters of baselines are set based on grid search.

For each dataset, the ratio of training, evaluation, and test set is 6 : 2 : 2. Each experiment is repeated 5 times, and the average performance is reported. We evaluate our method in two experiment scenarios: (1) In click-through rate (CTR) prediction, we apply the trained model to each piece of interactions in the test set and output the predicted click probability. We use *Accuracy* and *AUC* to evaluate the performance of CTR prediction. (2) In top- $K$  recommendation, we use the trained model to select  $K$  items with highest predicted click probability for each user in the test set, and choose *Precision@K*, *Recall@K*, *F 1@K* to evaluate the recommended sets.

We conduct an empirical study to investigate the correlation between the average number of common neighbors of an item pair in the KG and whether they have common rater(s) in RS. For each dataset, we first randomly sample one million item pairs, then count the average number of  $k$ -hop neighbors that the two items share in

the KG under the following two circumstances: (1) the two items have at least one common rater in RS; (2) the two items have no common rater in RS. The results are presented in Figures 4a, 4b, 4c, respectively, which clearly show that if two items have common rater(s) in RS, they likely share more common  $k$ -hop neighbors in the KG for fixed  $k$ . The above findings empirically demonstrate that *the similarity of proximity structures of two items in the KG could assist in measuring their relatedness in RS*. In addition, we plot the ratio of the two average numbers with different hops (i.e., dividing the higher bar by its immediate lower bar for each hop number) in Figure 4d, from which we observe that the proximity structures of two items under the two circumstances become more similar with the increase of the hop number. This is because any two items are probable to share a large amount of  $k$ -hop neighbors in the KG for a large  $k$ , even if there is no direct similarity between them in reality. The result motivates us to find a moderate hop number in RippleNet to explore users' potential interests as far as possible while avoiding introducing too much noise.

The results of all methods in CTR prediction and top- $K$  recommendation are presented in Table 3 and Figures 5, 6, 7, respectively.

Several observations stand out:

- CKE performs comparably poorly than other baselines, which is probably because we only have structural knowledge available, without visual and textual input.
- SHINE performs better in movie and book recommendation than news. This is because the 1-hop triples for news are too complicated when taken as profile input.
- DKN performs best in news recommendation compared with other baselines, but performs worst in movie and book recommendation. This is because movie and book names are too short and ambiguous to provide useful information.
- PER performs unsatisfactorily on movie and book recommendation because the user-defined meta-paths can hardly be optimal. In addition, it cannot be applied in news recommendation since the types of entities and relations involved in news are too complicated to pre-define meta-paths.
- RippleNet performs best among all methods in the three datasets. Specifically, RippleNet outperforms baselines by 2.0% to 40.6%, 2.5% to 17.4%, and 2.6% to 22.4% on *AUC* in movie, book, and news recommendation, respectively. RippleNet also achieves outstanding performance in top-*K* recommendation as shown in Figures 5, 6, and 7. Note that the performance of top-*K* recommendation is much lower for Bing-News because the number of news is significantly larger than movies and books.

In this paper, we propose RippleNet, an end-to-end framework that naturally incorporates the knowledge graph into recommender systems. RippleNet overcomes the limitations of existing embedding-based and path-based KG-aware recommendation methods by introducing preference propagation, which automatically propagates users' potential preferences and explores their hierarchical interests in the KG. RippleNet unifies the preference propagation with regularization of KGE in a Bayesian framework for click-through rate prediction. We conduct extensive experiments in three recommendation scenarios. The results demonstrate the significant superiority of RippleNet over strong baselines. For future work, we plan to (1) further investigate the methods of characterizing entity-relation interactions; (2) design non-uniform samplers during preference propagation to better explore users' potential interests and improve the performance.



## 附录 5 外文翻译

为了解决协同过滤的稀疏性和冷启动问题，研究人员通常会利用社交网络或项目属性等侧面信息来提高推荐性能。本文将知识图谱视为侧面信息的来源。为了解决现有的基于嵌入和基于路径的知识图谱感知推荐方法的局限性，我们提出了 RippleNet，一个将知识图谱自然融入推荐系统的端到端框架。RippleNet 类似于在水面上传播的实际涟漪，通过沿着知识图谱中的链接自动迭代地扩展用户的潜在兴趣，激发用户偏好在知识实体集合上的传播。用户历史上点击过的项目所激活的多个“涟漪”从而叠加形成用户对某个项目的偏好分布，可用于预测最终的点击概率。通过在真实世界数据集上的大量实验，我们证明了 RippleNet 在包括电影、书籍和新闻推荐在内的各种场景中都取得了显著的进步，超过了几种最先进的基线方法。

在线内容和服务的爆炸式增长为用户提供了大量选择，如新闻、电影、音乐、餐馆和书籍。推荐系统（RS）旨在通过为用户找到一小部分符合其个性化兴趣的项目来应对信息爆炸。在推荐策略中，协同过滤（CF）考虑了用户的历史交互，并根据他们潜在的共同偏好进行推荐，取得了巨大成功。然而，基于 CF 的方法通常存在用户-物品交互稀少和冷启动问题。为了解决这些局限性，研究人员提出将社交网络、用户/物品属性、图像和上下文等侧边信息纳入 CF。

在各种类型的辅助信息中，知识图谱（KG）通常包含更多有关物品的事实和联系。知识图谱是一种有向异构图，其中节点对应实体，边对应关系。最近，研究人员提出了一些学术知识图谱，如 NELL1 和 DBpedia2，以及商业知识图谱，如 Google Knowledge Graph3 和 Microsoft Satori4。这些知识图谱已成功应用于许多领域，如知识图谱补全、问题解答、词嵌入和文本分类。

受到在各种任务中成功应用 KG 的启发，研究人员也尝试利用 KG 来提高推荐系统的性能。如图 1 所示，KG 可以从三个方面为推荐带来益处：（1）KG 引入了条目间的语义关联，有助于发现它们之间的潜在联系，提高推荐条目的精确度；（2）KG 由各种类型的关系组成，有助于合理扩展用户的兴趣，增加推荐条目的多样性；（3）KG 将用户的历史记录和推荐记录联系起来，从而为推荐系统带来可解释性。一般来说，现有的 KG 感知推荐可分为两类。

第一类是基于嵌入的方法，这种方法使用知识图嵌入（KGE）算法对 KG 进行预处理，并将学习到的实体嵌入纳入推荐框架。例如，深度知识感知网络（Deep Knowledge-aware Network, DKN）将实体嵌入和词嵌入视为不同的通道，然后设计了一个 CNN 框架，将它们结合在一起进行新闻推荐。协作知识库嵌入（CKE）在统一的贝叶斯框架中将知识嵌入、文本嵌入和项目图像嵌入与 CF 模块相结合。Signed Heterogeneous Information Network Embedding（SHINE）设计了深度自动编码器来嵌入情感网络、社交网络和个人资料（知识）网络，用于名人推荐。基于嵌入的方法在利用 KG 辅助推荐系统方面表现出很高的灵活性，但这些方法所采用的 KGE 算法通常更适合图内应用（如链接预测）而非推荐，因此学习到的实体嵌入在表征项目间关系方面不够直观和有效。

第二类是基于路径的方法，这些方法探索 KG 中项目之间的各种连接模式，为推荐提供额外指导。例如，个性化实体推荐（PER）和基于元图的推荐（Meta-Graph Based Recommendation）将 KG 视为异构信息网络（HIN），并提取基于元路径/元图的潜在特征，以表示用户与项目之间沿着不同类型关系路径/图的连通性。基于路径的方法能以更自然、更直观的方式利用 KG，但它们在很大程度上依赖于人工设计的元路径，这在实践中很难优化。另一个问题是，在某些场景（如新闻推荐）中，实体和关系不在一个领域内，不可能设计出手工制作的元路径。

针对现有方法的局限性，我们提出了端到端知识图谱感知推荐框架 RippleNet。RippleNet 专为点击率（CTR）预测而设计，它将用户-项目对作为输入，并输出用户参与（如点击、浏览）项目的概率。RippleNet 背后的关键理念是偏好传播：对于每个用户，RippleNet 将其历史兴趣作为 KG 中的种子集，然后沿着 KG 链接迭代扩展用户的兴趣，以发现其对候选项目的分层潜在兴趣。我们将偏好传播类比为雨滴在水面上传播时产生的实际涟漪，其中多个“涟漪”叠加形成用户在知识图谱上的偏好分布结果。RippleNet 与现有文献的主要区别在于，RippleNet 综合了上述两类方法的优点：(1) RippleNet 通过偏好传播将知识图谱方法自然地融入到推荐中；(2) RippleNet 可以自动发现从用户历史记录中的项目到候选项目的可能路径，而无需任何人工设计。

实验中，我们将 RippleNet 应用于电影、书籍和新闻推荐这三个真实场景。实验结果表明，与最先进的推荐基线相比，RippleNet 在电影、书籍和新闻推荐方面的

AUC 分别提高了 2.0% 至 40.6%、2.5% 至 17.4% 和 2.6% 至 22.4%。我们还发现，RippleNet 从知识图谱的角度为推荐结果的可解释性提供了新的视角。

总之，在本文中的贡献如下：

- 据我们所知，这是第一项在知识图谱感知推荐中结合基于嵌入和基于路径的方法的工作。

- 我们提出了 RippleNet，一个利用知识图谱辅助推荐系统的端到端框架。RippleNet 通过迭代传播用户在 KG 中的偏好，自动发现用户的分层潜在兴趣。

- 我们在三个真实世界的推荐场景中进行了实验，结果证明 RippleNet 比几种最先进的基线方法更有效。

图谱感知推荐问题的表述如下。在一个典型的推荐系统中，让  $U = \{u_1, u_2, \dots\}$  和  $V = \{v_1, v_2, \dots\}$  分别表示用户集和项目集。用户-项目交互矩阵  $Y = \{y_{uv} | u \in U, v \in V\}$  是根据用户的隐式反馈定义的，其中，如果观察到交互  $(u, v)$ ，则  $y_{uv} = 1$ ；否则， $y_{uv} = 0$ 。

$y_{uv}$  值为 1 表示用户  $u$  与项目  $v$  之间存在隐式交互，如点击、观看、浏览等行为。除了交互矩阵  $Y$ ，我们还有一个知识图谱  $G$ ，它由大量实体相关-实体三元组  $(h, r, t)$  组成。这里， $h \in E$ 、 $r \in R$ 、 $t \in E$  分别表示知识三元组的头、关系和尾， $E$  和  $R$  表示知识图谱中实体和关系的集合。例如，三重（侏罗纪公园，film.film.director，史蒂文-斯皮尔伯格）表示史蒂文-斯皮尔伯格是电影《侏罗纪公园》的导演。例如，电影 "侏罗纪公园" 与 KG 中的同名实体相关联，而标题为 "法国熊猫宝宝首次公开亮相" 的新闻与实体 "法国" 和 "熊猫" 相关联。

在给定互动矩阵  $Y$  和知识图谱  $G$  的情况下，我们的目标是预测用户  $u$  是否对之前没有互动的项目  $v$  有潜在兴趣。我们的目标是学习一个预测函数  $y^{uv} = F(u, v; \Theta)$ ，其中  $y^{uv}$  表示用户  $u$  点击商品  $v$  的概率， $\Theta$  表示函数  $F$  的模型参数。

在本节中，我们将对 RippleNet 在电影、书籍和新闻推荐 5 这三个真实世界场景中的表现进行评估。我们首先介绍数据集、基线和实验设置，然后介绍实验结果。在本节中，我们还将给出一个可视化案例研究，并讨论超参数的选择。

由于 MovieLens-1M 和 Book-Crossing 是显式反馈数据，我们将其转换为隐式反馈，即每个条目都标记为 1，表示用户已对该条目进行了评分（MovieLens-1M 的评分阈值为 4，而 Book-Crossing 由于其稀疏性没有设置阈值），并对每个用户标记

为 0 的未评分集进行采样，该集与已评分集的大小相等。对于 MovieLens-1M 和 Book-Crossing，我们使用用户和条目的 ID 嵌入作为原始输入，而对于 Bing-News，我们将新闻条目的 ID 嵌入和标题的平均单词嵌入作为条目的原始输入，因为新闻标题通常比电影或书籍名称长很多，因此能为推荐提供更有用的信息。

我们使用 Microsoft Satori 为每个数据集构建知识图谱。对于 MovieLens-1M 和 Book-Crossing，我们首先从整个知识图谱中选取关系名包含 "电影" 或 "书籍" 且置信度大于 0.9 的三元组子集。给定子 KG，我们通过将电影/书籍的名称与三元组的尾部 (`head, film.film.name, tail`) 或 (`head, book.book.title, tail`) 进行匹配，收集所有有效电影/书籍的 ID。为简单起见，没有匹配实体或有多个匹配实体的项目将被排除在外。然后，我们将 ID 与所有 KG 三元组的头部和尾部进行匹配，从子 KG 中选择所有匹配良好的三元组，并将实体集迭代扩展到四跳。Bing-News 的构建过程与此类似，但有以下不同：(1) 我们使用实体链接工具提取新闻标题中的实体；(2) 由于新闻标题中的实体不属于某一特定领域，因此我们对关系名称不加限制。

我们将所提出的 RippleNet 与下列最新基准进行了比较：

- CKE [43] 将 CF 与结构知识、文本知识和视觉知识结合在一个统一的推荐框架中。我们在本文中将 CKE 作为 CF 加结构知识模块来实现。

- SHINE [32] 设计了深度自动编码器来嵌入情感网络、社交网络和档案（知识）网络，用于名人推荐。在这里，我们使用自动编码器来处理用户与项目的交互，并使用项目简介来预测点击概率。

- DKN [33] 将实体嵌入和单词嵌入视为多通道，并将它们结合到 CNN 中，用于点击率预测。本文使用电影/书籍名称和新闻标题作为 DKN 的文本输入。

- PER [42] 将 KG 视为 HIN，并提取基于元路径的特征来表示用户与项目之间的连接性。在本文中，我们使用 PER 的所有条目-属性-条目特征（如 "电影-导演-电影"）。

- LibFM [20] 是 CTR 场景中广泛使用的基于特征的因式分解模型。我们将用户 ID、项目 ID 和从 TransR [14] 中学习到的相应平均实体嵌入作为 LibFM 的输入。

- Wide&Deep[6]是一种通用的深度推荐模型，结合了（宽）线性通道和（深）非线性通道。与 LibFM 类似，我们也使用用户、项目和实体的嵌入来为 Wide&Deep 提供输入。

在 RippleNet 中，我们为 MovieLens-1M/BookCrossing 设置了跳数  $H = 2$ ，为 Bing-News 设置了跳数  $H = 3$ 。实验结果表明，跳数越大，性能越差，但计算开销越大。

其中， $d$  表示项目和知识图谱的嵌入维度， $\eta$  表示学习率。超参数是通过优化验证集上的 AUC 确定的。为公平起见，所有比较基线的潜在维度设置与表 2 相同。

每个数据集的训练集、评估集和测试集的比例为 6:2:2。每个实验重复 5 次，并报告平均性能。我们在两个实验场景中评估了我们的方法：(1) 在点击率 (CTR) 预测中，我们将训练好的模型应用于测试集中的每条互动信息，并输出预测的点击概率。我们使用准确率和 AUC 来评估点击率预测的性能。(2) 在 Top-K 推荐中，我们使用训练有素的模型为测试集中的每个用户选择  $K$  个预测点击概率最高的项目，并选择 Precision@K、Recall@K、F 1@K 来评估推荐集。

在点击率预测和 Top-K 推荐方面的结果中，其中有几点值得注意：

- CKE 的表现比其他基线方法差，这可能是因为我们只有结构知识，而没有视觉和文本输入。

- SHINE 在电影和书籍推荐方面的表现要好于新闻推荐。这是因为新闻的 1 跳三元组在作为轮廓输入时过于复杂。

- PER 在电影和图书推荐方面的表现并不令人满意，因为用户定义的元路径很难达到最优。此外，它也无法应用于新闻推荐，因为新闻中涉及的实体类型和关系过于复杂，无法预先定义元路径。

- 作为两款通用推荐工具，LibFM 和 Wide&Deep 的性能令人满意，这表明它们可以在算法中充分利用 KG 的知识。

- 在三个数据集中，RippleNet 在所有方法中表现最佳。具体来说，在电影、书籍和新闻推荐方面，RippleNet 的 AUC 分别比基线高出 2.0% 到 40.6%、2.5% 到 17.4% 和 2.6% 到 22.4%。如图 5、图 6 和图 7 所示，RippleNet 在 top-K 推荐方面也取得了出色的表现。需要注意的是，Bing-News 的 top-K 推荐性能要低得多，因为新闻的数量明显多于电影和书籍。

本文提出的 RippleNet 是一个端到端框架，可将知识图谱自然融入推荐系统。 RippleNet 克服了现有的基于嵌入和基于路径的知识图谱感知推荐方法的局限性，它引入了偏好传播，自动传播用户的潜在偏好并探索他们在知识图谱中的分层兴趣。 RippleNet 将偏好传播与 KGE 正则化统一在贝叶斯框架中，用于预测点击率。我们

在三个推荐场景中进行了广泛的实验。结果表明，RippleNet 明显优于强基线。在未来的工作中，我们计划：（1）进一步研究描述实体-关系交互的方法；（2）在偏好传播过程中设计非均匀采样器，以更好地发掘用户的潜在兴趣并提高性能。

# 燕山大学毕业设计（论文）指导教师意见书

学院：信息科学与工程学院

专业(方向)：计算机科学与技术

题目名称		基于知识图谱的饮食推荐系统的设计与实现				
学号		202011040034	姓名	王晓哲	班级	20-2 班
序号	评审项目	指标			满分	评分
1	项目过程的管理意识 5分	(2分) 1.1 能独立查阅文献和调研；能较好地写出开题报告和中期报告，项目实施过程中各要素组织基本齐全、合理； (3分) 1.2 有综合、收集和正确利用各种信息及获取新知识的能力，能考虑非技术指标因素。			5	5
2	自我学习意识和工作态度 5分	(2分) 2.1 参加了毕业设计（论文）的各个训练环节，按期独立完成规定的工作任务，可以很好的综合运用所学专业知识解决遇到的问题；能够独立查阅相关资料内容，可以很好的解决实际问题。 (3分) 2.2 遵守纪律，学习认真；尊敬指导教师，能够听取指导教师的意见和指导；不存在学术不端行为。			5	5
是否同意参加答辩： <input checked="" type="checkbox"/> 同意 <input type="checkbox"/> 不同意					总分	10
评语：   指导教师签名：王晓哲 日期：2024.5.30						

# 燕山大学毕业设计（论文）评审表（评阅人用）

学院：信息科学与工程学院

专业（方向）：计算机科学与技术

题目名称		基于知识图谱的饮食推荐系统的设计与实现				
学号	202011040034	姓名	王晓哲	班级	20-2 班	
序号	评审项目	指标			满分	评分
1	报告和论文的撰写质量 10分	(5分) 1.1 逻辑结构。毕业设计（论文）主题基本明确、结构基本合理、有一定层次性，引用资料基本准确，附录材料与毕业设计（论文）基本相符。 (5分) 1.2 写作能力。毕业设计（论文）语言基本流畅，整体内容能够体现设计主题；文字表述、语法应用、书写格式、图表注释、缩略词等基本符合国家相关标准；参考文献标注基本规范、来源清楚。			10	9
2	外文资料阅读与翻译 5分	(2分) 2.1 阅读能力。外文原文与毕业题目相关，来源于计算机领域期刊或者会议、不少于6页。 (3分) 2.2 翻译能力。中文翻译与外文原文对应，翻译准确、流畅。			5	4
3	创新和发展意识 5分	(2分) 3.1 创新意识。设计能够体现一定的创新思路。 (3分) 3.2 发展意识。设计方案正确，设计方法得当，体现一定分析解决本专业实际问题的综合设计能力。			5	7
					总分	18

评语：

通过

评阅人签名：

日期：2024.6.2

# 燕山大学毕业设计（论文）答辩评审表

学院：信息科学与工程学院

专业（方向）：计算机科学与技术

题目名称		基于知识图谱的饮食推荐系统的设计与实现				
学号	202011040034	姓名	王晓哲	班级	20-2 班	
指导教师姓名		陈子军	职称	教授		
序号	评审项目	指标			满 分	评 分
1	开发工具运用能力 10 分	(5分) 1.1 设计能力。可以很好的应用开发工具设计符合国家或行业标准的毕业设计（论文），体现一定分析解决本专业实际问题的综合设计能力，基本达到选题的目标要求。 (5分) 1.2 系统能力。体现出较为扎实的专业基本知识，基本能够综合运用开发工具解决毕业设计（论文）相关问题。软件系统能够正常运行，数据正确，代码命名规范、缩进风格统一，能够准确的无误的讲解代码的运行逻辑。			10	10
2	工作总结和成果展示 10 分	(5分) 2.1 工作总结。答辩表述清楚，语言基本流畅，整体内容能够体现毕业设计（论文）主题；体现出较为扎实的专业基本知识，核心概念基本明确。 (5分) 2.2 成果展示。态度认真，对完成的毕业设计（论文）内容完全掌握；回答问题有理有据，基本概念清楚，主要问题回答准确，有一定深度。			10	10
					总分	20
<b>评语：</b> 						
答辩委员会小组成员： 姓名职称（签名）：何沛淳 教授 姓名职称（签名）： 姓名职称（签名）：任洪伟 讲师 姓名职称（签名）： 姓名职称（签名）：胡立军 副师 姓名职称（签名）： 答辩委员会（小组）负责人签名：何沛淳 2024年6月2日						

## 燕山大学毕业设计（论文）评分表

### 开题考核：

开题考核（满分 20 分）			
当前社会发展现状的认识 (5 分)	专业领域技术发展现状的调研 (5 分)	问题分析和解决方案 (10 分)	开题成绩

### 中期考核：

中期考核（满分 30 分）			
设计方案的可行性分析 (8 分)	问题研究能力的评价 (10 分)	系统设计能力的考核 (12 分)	中期成绩

## 燕山大学毕业设计（论文）评分表

指导教师评分：

导师评分（满分 10 分）		
项目过程的管理意识 (5 分)	自我学习意识和工作态度 (5 分)	导师评分

指导教师签字：\_\_\_\_\_ 年 月 日

毕业论文答辩：（满分 40 分）

报告和论文的撰写质量 (10 分)	外文资料阅读与翻译 (5 分)	创新和发展意识 (5 分)

开发工具运用能力 (10 分)	工作总结和成果展示 (10 分)	答辩成绩

答辩组组长签字：\_\_\_\_\_ 年 月 日

综合其开题成绩、中期成绩、导师评分、答辩成绩，

该本科生毕业设计(论文)的总成绩为：\_\_\_\_\_。

(A+    A    B+    B    C+    C    D+    D    F)