Intelligent Cognitive Systems for Automated Recipe Generation

By

Muhammad Saad Razzaq

A thesis submitted to the Computer Science & Information Technology.

University of Sargodha - Pakistan

In partial fulfillment of the requirement for the degree of

PhD (Computer Science)

Supervisor

Dr. Muhammad Ilyas
Assistant Professor of Computer Science & Information Technology
University of Sargodha

Submission Date: February 21, 2024

Declaration

I Muhammad Saad Razzaq (SCSF17E005) student of the Computer Science & Information Technology, University of Sargodha hereby declare that I have produced the work presented in this thesis, during the scheduled period of study. I also declare that I have not taken any material from any source except referred to wherever due that amount of plagiarism is within acceptable range. If a violation of HEC rules on research has occurred in this thesis, I shall be liable to punishable action under the plagiarism rules of the HEC.

Date: February 21, 2024 Signature: .

Muhammad Saad Razzaq

SCSF17E005

DEDICATION

I would like to dedicate my thesis to my parents . . .

ACKNOWLEDGEMENTS

I would like to pay my gratitude to Allah Almighty that helps me and give me the courage to complete this thesis. I would also like to thanks my supervisors Dr. Muhammad Ilyas and Co-Supervisor Dr. Hajira Jabeen on there vigorous support through out the thesis time span. With out their support it would not be possible for me to complete this thesis. I would also thanks to my best friend and colleague Mr. Fahad Maqbool in his support and guidance through the degree time. I would also like to thanks my parents whom prayers would able me to complete this thesis.

Muhammad Saad Razzaq

SCSF17E005

THESIS SUBMISSION CERTIFICATE

It is certified that the research thesis entitles "**Intelligent Cognitive Systems for Automated Recipe Generation**" that has been submitted by **Muhammad Saad Razzaq** is an original research work that is not copied from or submitted to any other university for any degree requirements. The candidate has compiled the thesis according to the University of Sargodha plagiarism guidelines.

Dr. Muhammad Ilyas (Supervisor I)
Assistant Professor
University of Sargodha
Sargodha, Pakistan

Dr. Hajira Jabeen (Supervisor II)
Lead Big Data Team
GESIS-Leibniz Institute for the Social Sciences
Cologne, Germany

Incharge/Chairman
Dept. of Computer Science & IT
University of Sargodha

ABSTRACT

The semantic web is about adding semantics to otherwise flat data to make it machine-understandable. However, the enrichment of this semantic information in data for computational creativity has remained under explored. Machine-generated recipes have gained attention in the last few years due to the increasing demand of personalized recipes from food enthusiasts, sick persons, and nutrient-conscious people. Recipe is an art as well a science, therefore machine-understandabiliy of recipes is the key ingredient to create valid machine-generated recipes. Moreover, recipe is a complex process that also needs a large scale collection of recipe dataset with plurality of attributes to understand and model the recipe process.

In this thesis we covers creativity in culinary arts by adding semantics to the recipes, not only making recipes machine-understandable but also assisting machines in generating new recipes. From the user perspective, the demand for personalized recipes is increasing. This includes specialized food for sick (e.g, allergies), diet conscious, or food enthusiasts. Therefore, it is important to explore the semantics underlying recipes to develop computer assisted recipes. In this thesis, we propose the recipe ontology *RecipeOn*. The purpose of the development of this ontology is, on one hand assisting machines in understanding the semantics of recipes, and on the other hand encoding the information crucial for personalized recipe generation. This includes e.g. nutritional information, ingredient-action relations, ingredient-diet relations, ingredient substitutions and cooking time. In addition, we are targeting the computational intelligence task of new recipe generation, where the machine-readable, semantic-aware recipes are recombined to generate new recipes using alternative ingredients and actions. RecipeOn model not only the recipe knowledge but also the sequence of activities. Each recipe has at least one ingredient and each ingredient has at least one action performed on it. seq:directlyPrecedes and seq:directlyFollows are used to specify the order of the activities. *RecipeOn* also defines the concept of procedures that groups actions and ingredients to complete a sub task of a recipe. Many knowledge graphs have been created in the food domain that focus on pairing, chemical composition, and nutritional information of ingredients to generate novel recipes. Semantic-aware recipes demand more than just a knowledge-base of ingredients and instructions. Ingredient-ingredient, ingredient-action, and action-action relationships need to be tailored to understand the recipe process. After developing RecipeOn ontology we have developed a knowledge graph

RecipeKG using 0.8M-Recipes based on the concepts and relationships defined by RecipeOn[1] ontology. *RecipeKG* comprises 52.96 million instances and 209 million facts. *RecipeKG* is developed by the dataset contained 0.8 Million Recipes ($0.8M - Recipes$) compiled from top ranked (based on internet traffic) recipe websites.

Generative AI e.g. Large Language Models (LLMs) can be used to generate new recipes. However, LLMs struggle with more complex aspects like recipe semantics and process comprehension. Furthermore, LLMs have limited ability to account for user preferences without hallucinations since they are based on statistical patterns. As a result, these recipes may be invalid. Evolutionary algorithms inspired by the process of natural selection are optimization algorithms that use stochastic operators to generate new solutions. Evolutionary algorithms can generate large number of solutions from the set of possible solution space. Moreover, these algorithms have the capability to incorporate user preferences in fitness function to generate novel recipes that are more aligned with the fitness objective.

Finally In this thesis, we propose the *EvoRecipes* framework to generate novel recipes. The *EvoRecipes* framework utilizes both Genetic Algorithm and generative AI in addition to *RecipeOn* ontology, and *RecipeKG* knowledge graph. Genetic Algorithms explore the large solution space of encoded recipe solutions and are capable of incorporating user preferences, while LLMs are used to generate recipe text from encoded recipe solutions. *EvoRecipes* uses a population of context-aware recipe solutions from the *RecipeKG* knowledge graph. *RecipeKG* encodes recipes in RDF format using classes and properties as defined in the *RecipeOn* ontology. Moreover, to evaluate the alignment of *EvoRecipe* generated recipes with multiple intended objectives, we propose a fitness function that can incorporate multiple objectives. Additionally, to evaluate the quality of the *EvoRecipe* generated recipes while considering the subjective nature of recipes, we conducted a survey using multi-dimensional metrics (i.e., contextual, procedural, novelty) to assess the effectiveness of *EvoRecipes*. Results show that EvoRecipes are novel, can incorporate user preferences and valid.

Resource:

https://hajirajabeen.github.io/RecipeKG/0.8M-Recipes/

https://hajirajabeen.github.io/RecipeKG

[1] https://hajirajabeen.github.io/EvoRecipesOntology/

Contents

Declaration i

Dedication ii

Acknowledgements iii

Thesis Submission Certificate iv

Abstract v

List of Figures x

List of Tables xiii

1 Introduction 1
- 1.1 Motivation 4
- 1.2 Problem Statement 6
- 1.3 Research Questions 7
- 1.4 Thesis Contribution 9
  - 1.4.1 List of Publications 10
- 1.5 Thesis Structure 11

2 Preliminaries 12
- 2.1 Food & Culinary Recipes 12
- 2.2 Semantic Representation & Technologies 13
  - 2.2.1 Resource Document Framework (RDF) 13
  - 2.2.2 Ontologies 14
  - 2.2.3 SPARQL 14
  - 2.2.4 Knowledge Graph 15
- 2.3 Computational Creativity & Evolutionary Algorithms 16
  - 2.3.1 Computational Creativity in Culinary Recipes 17
  - 2.3.2 Genetic Algorithms 18

3 Related Work 20
- 3.1 Food Ontologies 20
- 3.2 Recipe Datasets and Food Knowledge Graphs 22
- 3.3 Recipe Evolution 26

4 RecipeOn: An Ontology to Represent Recipe as a Process 30
- 4.1 Development of RecipeOn Ontology 32
  - 4.1.1 Requirements of RecipeOn 34
    - 4.1.1.1 Functional Requirements of RecipeOn 34
    - 4.1.1.2 Non-Functional Requirements of RecipeOn 37
- 4.2 Ontology Overview 38
  - 4.2.1 Core-Recipe 39

4.2.2 Ingredients . . . 42
4.2.3 Actions . . . 48
4.2.4 Nutrition . . . 52
4.2.5 Procedures . . . 56
4.2.6 Property Restrictions . . . 58
4.3 Ontology Evaluation . . . 61
4.3.1 OOPS Evaluation . . . 62
4.3.2 Ontology Quality . . . 62
4.3.2.1 Accuracy . . . 62
4.3.2.2 Adaptability . . . 62
4.3.2.3 Clarity . . . 63
4.3.2.4 Completeness . . . 63
4.3.2.5 Consistency . . . 63
4.3.3 Ontology Metrics . . . 63
4.4 RecipeOn Example . . . 65
4.5 New Recipe Generation: The Use Case of RecipeOn . . . 65
4.5.1 Alternative Ingredient/Action . . . 66
4.5.2 Alternative Procedure . . . 67
4.6 Summary . . . 69

5 RecipeKG: Knowledge Graph to Manifest Culinary Recipes 71
5.1 0.8 Million Recipes (0.8M-Recipes) . . . 73
5.1.1 Data Extraction . . . 74
5.1.2 Data Consolidation . . . 76
5.1.3 Data Preprocessing . . . 76
5.1.4 Data Transformation . . . 78
5.1.4.1 Ingredients Transformation . . . 78
5.1.4.2 Instructions Transformation . . . 79
5.1.5 Dataset Entities . . . 80
5.1.5.1 Entity: CoreRecipe . . . 82
5.1.5.2 Entity: Ingredients . . . 83
5.1.5.3 Entity: Instructions . . . 86
5.1.5.4 Entity: Nutrition . . . 89
5.2 RecipeKG: A Semantically-Driven Knowledge Graph to Manifest Culinary Recipes . . . 90
5.3 Use cases and Discussion . . . 98
5.3.1 Recipe Profile . . . 98
5.3.2 Recipes for Sick and Diet Conscious People . . . 99
5.4 Summary . . . 100

6 EvoRecipes: A Generative Approach for Evolving Context-Aware Recipes 101
6.1 Preliminaries . . . 104
6.1.1 RecipeOn . . . 104
6.1.2 RecipeKG . . . 104
6.2 EvoRecipes: A Generative Approach for Evolving Context-Aware Recipes . . . 106
6.2.1 Initial Population . . . 107

6.2.2 Selection . . . 107
6.2.3 Mutation . . . 108
6.2.3.1 Ingredient Substitution . . . 109
6.2.3.2 Action Substitution . . . 109
6.2.3.3 Action Interchange . . . 110
6.2.4 Crossover: Procedure Substitution . . . 112
6.2.5 Quantitative Evaluation of Recipe . . . 114
6.2.5.1 Novelty . . . 115
6.2.5.2 Simplicity . . . 116
6.2.5.3 Visual Appeal . . . 116
6.2.5.4 Feasibility . . . 117
6.2.5.5 Fitness Function of a Recipe (f) . . . 117
6.2.6 Recipe RDF to Recipe Text using Generative AI . . . 118
6.3 Qualitative Recipe Evaluation . . . 119
6.3.1 Contextual Metrics . . . 120
6.3.2 Procedural Metrics . . . 121
6.3.3 Novelty Metrics . . . 122
6.3.4 Survey and Analysis . . . 123
6.4 Result and Discussion . . . 125
6.4.1 Experimental Setup . . . 125
6.4.2 Experiments . . . 126
6.4.3 Comparison with Recipe Evolution Techniques . . . 128
6.5 Summary . . . 129

7 Conclusion & Future Directions **130**

Bibliography **132**

List of Figures

1.1 Google Trend: Recipe vs Food near me. 2

2.1 An RDF Graph of Fluffy Microwave Scrambled Eggs Recipe 16

4.1 Classes and Properties Involved in Core-Recipe Module. 39
4.2 Ingredients Class Hierarchy 43
4.3 Object Properties and Classes related to Ingredient Module. . . . 44
4.4 Object Properties and Classes Related to Action Module. 49
4.5 Hierarchy of Classes, Sub-Classes, and Individuals in the Action Module. 50
4.6 Object Properties and Classes related to Nutrition Module. 53
4.7 Object Properties and Classes Related to Procedures Module. . . . 57
4.8 Recipe 59
4.9 Person 59
4.10 AggregateRating 59
4.11 Excerpt of Ingredient Class 60
4.12 Excerpt of Action Class. 60
4.13 Nutrient 60
4.14 Quantity 61
4.15 QuantityValue 61
4.16 Unit 61
4.17 Excerpt of Procedure Class. 61
4.18 Knowledge graph of "Roast Beef Dip Sandwich with Herbed Garlic Au Jus" Recipe 66
4.19 Rice Recipe with Alternative Procedure 68

5.1 Recipes Extracted from Top 50 Recipe Websites 75
5.2 POS Tagging of Ingredient Phrases Using the Spacy Library. In the First Ingredient Phrase "400 g" is the Size and "tins" is the Unit. These both have been Misidentified. In the second phrase "g" has also been missed which should be Mapped to Unit. 78
5.3 POS Tagging of Ingredient Phrases using Stanford Core-Nlp Library. In the first Ingredient Phrase "x" has been Misidentified as a unit. In the Second Phrase "g" has been Missed. 79
5.4 Identification of Quantity, Unit, Size, and Ingredient using Regular Expressions. In both Ingredient Phrases Attributes have been Correctly Identified. 79
5.5 POS Tagging of Instruction Statements using the Spacy Library. . 80
5.6 POS Tagging of Instruction Statements using Stanford Core-NLP Library. 80
5.7 Ingredients and Actions Extraction from Instruction Statement using Hybrid Approach 80
5.8 CoreRecipe has 1-M Relationship with Ingredients and Instructions, while 1-1 Relationship with Nutrition. 82

5.9 Average Aggregate Rating of each Recipe Website Against the Total Number of Aggregate Ratings. The Color of the Sphere Shows the Ranking of the Website Based on Traffic Volume while the Size of the Sphere Shows the Volume of the Recipes on the Website. . . . 83
5.10 Bipartite Graph representing (most frequently occurring) Ingredients as Nodes having a Relationship with other (most frequently occurring) Ingredients based on Co-Existence in the same Recipe. 84
5.11 Five Frequently Used Starting Recipe Actions along with Frequently used following Actions. 87
5.12 Correlation Matrix Between 25 Frequently used Ingredients and 25 Frequently used Starting Actions. Higher Color Intensity Represents strong Co-Relation 88
5.13 Average Level of Calories in Different Cuisines. 89
5.14 *RecipeKG* Pipeline Comprising of Three Stages 93
5.15 Level-1 Nodes and Relationships of a Recipe in *RecipeKG*. The Recipe is the Root Node while all other Nodes are Level 1 Nodes with Corresponding Relationships from the Recipe Node. 94
5.16 Level-2 Nodes and Relationships of a Recipe in *RecipeKG*. Level-2 Ingredient Nodes in Yellow Color are Linked from the Recipe Node using a Blank Node with hasIngredient Relationship. 94
5.17 Level-3+ Action Nodes with hasAction and directlyPrecedes Relationships. 96

6.1 Caption for EOL 105
6.2 EvoRecipes framework generates new semantic-aware recipes (encoded in RDF format) using evolutionary operators and generates human-readable recipe text using the Davinci model of OpenAI GPT API 105
6.3 Alternative Ingredient Against Four Ingredient Classes (i.e. Rice, Barley, Corn, and Grain Types) 106
6.4 Fluffy Microwave Scrambled Eggs Available at AllRecipes 108
6.5 Mutated (Milk Substitution with Cheese) Fluffy Microwave Scrambled Eggs 108
6.6 Noodle Bowl Recipe available at AllRecipes 110
6.7 Mutated (action substitution & action interchanged) Noodle Bowl Formula 110
6.8 Categorization of Preparatory Actions for Alternative Action Mutation) 111
6.9 Categorization of Cooking Actions for Alternative Action (Mutation) 112
6.10 Categorization of Post-Cooking Actions for Alternative Action (Mutation) 112
6.11 caption for LOF 113
6.12 caption for LOF 114
6.13 Novel offspring Recipe Comprising two Procedures ("Unagi Sauce" and "Prepare the Salmon"). Ingredients and Actions involved in "Unagi Sauce" are shown using Grey-Colored Nodes while "Prepare the Salmon" is Represented using Green-Colored Nodes. 114
6.14 Average Novelty Over 10 Runs of *EvoRecipes* 125
6.15 Average Simplicity Over 10 Runs of *EvoRecipes* 126

6.16 Average Visual Appeal Over 10 Runs of *EvoRecipes* 127
6.17 Average Fitness Value Over 10 Runs of *EvoRecipes* 127

List of Tables

3.1 Recipe Datasets with Number of Records and Attributes 23

4.1 RecipeOn' Ontology Requirements Specification Document (ORSD). 32
4.2 Re usability of Classes, Object properties, and Data properties from various Namespaces to RecipeOn Ontology. 38
4.3 SPARQL Queries against Competency Questions Q_c1 mentioned in Table 4.1. 40
4.4 SPARQL Queries against Competency Questions Q_c2 mentioned in Table 4.1. 45
4.5 SPARQL Queries against Competency Questions Q_c3 mentioned in Table 4.1. 51
4.6 SPARQL Queries against Competency Questions Q_c4 mentioned in Table 4.1. 53
4.7 SPARQL Queries against Competency Questions Q_c5 mentioned in Table 4.1. 57
4.8 Ontology Metrics. 64
4.9 Ingredients and Alternative Ingredients 67
4.10 Actions and Alternative Actions 69

5.1 Sample Data from 0.8M-Recipes and Corresponding Operations for Data Cleansing 77
5.2 Statistics Summary of Ingredient Phrases Transformation using Stanford Core-NLP, Spacy, and Regular Expressions. 80
5.3 Statistics Summary of Instruction Statements Transformation using Stanford Core-NLP, Spacy, and Hybrid Approach. 81
5.4 Descriptive Statistics of $RecipeKG$ Entities 92
5.5 Number of Facts in $RecipeKG$ against various Semantic Relationships (as defined by $RecipeOn$) of Core Recipe Module. 95
5.6 Number of Facts in $RecipeKG$ against various Semantic Relationships (as defined by $RecipeOn$) of Ingredient Module. 95
5.7 Number of Facts in $RecipeKG$ against arious Semantic Relationships (as defined by $RecipeOn$) of action module. 97
5.8 Number of Facts in $RecipeKG$ against various Semantic Relationships (as defined by $RecipeOn$) of Procedure Module. 97
5.9 Number of facts in $RecipeKG$ against various semantic relationships (as defined by $RecipeOn$) of nutrition module. 97

6.1 Recipe Evaluation Comparing Human-generated Recipe with EvoRecipe Generated Recipe. 123

Chapter 1

Introduction

Cooking food is one of the key indicators of civilized life that also sets humans apart from other animals. With time, different cultures have developed different tastes, staple foods, and cuisines, based on their cultural and geographical differences. With the advancements in globalization and the availability of most of ingredients across the globe, one can easily experiment with novel and specialized recipes to suit individual diet preferences. However, the creation of recipes is an art that combines food, science, and culinary experience with expertise. While culinary experts frequently develop innovative recipes, it is known that humans use a tiny proportion of possible food combinations in their diets (Ahn and Ahnert, 2013). It is a challenging task for humans to find the optimal food combinations that can satisfy nutritional needs while satisfying the palate. The diversity of food and cuisine has intrigued a range of research areas including anthropologists, psychologists, sociologists, and chemists. While a few models (Ahn et al., 2011, De Klepper, 2011, Jabeen et al., 2020, Jain et al., 2015) have been created to depict the recipe evolution, the rules defining and quantifying the principals of cooking, like ingredient pairing, or ingredient preparation, remain far from understood. Analogous to any other art, defining rules for all aspects of culinary creativity may not be possible. Nonetheless, semantic modeling of these concepts can help in generating specialized recipes.

A cooking recipe is a process that contains a set of directions and a list of ingredients for preparing a food dish. Although, there are various recipe search websites and blogs, but they do not allow searching based on useful key factors like the experience level of the cook, ingredients, availability, or diet-related constraints. Therefore, a relatively simple recipe might become a challenge for new cooks. Recently, due to Covid-19-related closures, it is assumed that cooking at home has become more prevalent. According to Google search trends,

FIGURE 1.1: Google Trend: Recipe vs Food near me.

the keyword "Recipe" has been searched at least 4 times more than "Food near me" worldwide as shown in Figure 1.1.

While a multitude of recipe-related ontologies exists (e.g, An ontology for menu planning system (Snae and Bruckner, 2008), Food ontology using word embedding technique (Youn et al., 2020), ISO-Food for isotopic data in Food Science (Eftimov et al., 2019), and Bionutrition ontology based on diet plans (Musen et al., 2012b)), these ontologies do not provide the necessary classes and relationships to represent recipes as a cooking process. Furthermore, they do not usually cover the requirements of a specialized recipe search, or assist in building an autonomous system. Although Ontology Design Patterns for cooking recipes (Sam et al., 2014) claims to represent the recipe as a process, it lacks the sequence of actions, action-ingredient relationship, and action-action relationships, where actions are preparatory actions (i.e. washing, cutting, dicing), cooking actions (i.e. boiling, baking, mixing) and post cooking actions(i.e. storing, cooling, finishing). In this thesis, we have developed a recipe ontology *RecipeOn* that adds creativity to culinary art by adding semantics to the recipes. *RecipeOn* models the recipe knowledge along with the sequence of actions. It also helps to generate novel recipes using alternative ingredients and actions.

Culinary cooking is a complex task dependent on many factors (selection of ingredients, proportions of ingredients, substitution of ingredients, actions on ingredients, sequence of actions, and relationship between actions). Food experts say that the simplest recipes (e.g. boiled rice, baked chicken, french fries) are often difficult to cook as there is less number of choices to hide behind fancy items (i.e. sauces, sprinkles, garnish). This further increases the significance of the above-mentioned factors. Due to the complexity of the cooking process, learning culinary recipes requires a large scale collection of recipes (like Recipe1M+[1] (Marin et al., 2019)). In recent years, the volume of the recipe datasets has increased from a few thousands (Bossard et al., 2014, Chen et al., 2009, Jain, 2020) to million plus items (Harashima et al., 2016, 2017, Salvador et al., 2017). However, less attention has been paid to rich recipe features (i.e. relevant data and metadata) extraction. Understanding culinary recipes demands rich metadata. This may include descriptive metadata (like published date, author, cooking time, title, etc) or contextual metadata (utensils used during an action).

Numerous application areas of the food domain demand exploration of food in a multitude of dimensions (i.e. recipe representation (Li and Zaki, 2020, Pallagani et al., 2022), recipe generation (Agarwal et al., 2009, Antô et al., 2020, Draschner et al., 2019, Jabeen et al., 2019, 2020, Naik and Polamreddi, 2015, Varshney et al., 2019), food creativity & assessment (Engisch, 2020, Jimenez-Mavillard and Suarez, 2022, Pinel and Varshney, 2014, Sakib et al., 2022), food management (Agarwal, 2020), ingredient substitution (Gim et al., 2022, Lu et al., 2020, Nadee and Unankard, 2021, Pan et al., 2020, Shirai et al., 2017), food recommendation (Beijbom et al., 2015, Elsweiler et al., 2017, Gim et al., 2021, Khilji et al., 2021, Lei et al., 2021), image to food retrieval (Bolaños et al., 2017, Chen and Ngo, 2016, Chen et al., 2020, 2018), food-based resource (Popovski et al., 2019), and food matching (Arffa et al., 2016, Su et al., 2014). RecipeDB[2] (Batra et al., 2020) has been developed to facilitate basic and advanced search based on ingredients, cuisines, flavor, plus along with 1M+ user interactions. In this thesis, we have developed a recipe dataset that contains around 0.8 million recipes[3], represents each recipe using a rich feature set of 50+ attributes, and is available in SQL format. Also, we have proposed a Recipe Knowledge Graph ($RecipeKG$) that is based on the semantic concepts and relationships from $RecipeOn$ ontology. $RecipeKG$ is semantically rich in recipe representation compared to its predecessors in the food domain. It comprises information about recipe features, a plurality of recipe metadata, provenance, nutrition, ingredients, actions on ingredients, successor actions after

[1] http://pic2recipe.csail.mit.edu/
[2] https://cosylab.iiitd.edu.in/recipedb
[3] https://github.com/HajiraJabeen/RecipeKG/tree/main/0.8M-Recipes

an action, predecessor actions of an action, and procedures comprising ingredients and actions. *RecipeKG* is a large scale semantic-aware recipe knowledge graph that is created to increase the machine-understandably of the recipe process. It contains 0.8 million recipe nodes, 8.9 million ingredient nodes, 10.8 million action nodes, and 1 million nutrient nodes.

Food is also considered a computational artifact (Deng et al., 2022) and researchers are exploring the potential of computational creativity to generate novel recipes (Antô et al., 2020, Jabeen et al., 2020, Loughran and O'Neill, 2017). Moreover, with the evolution of the semantic web, machine-understandable recipes have gained more attention in recent years. Several food ontologies (e.g. FoodOn[4] (Dooley et al., 2018), RecipeOn[5]) and knowledge graphs (e.g. FlavorGraph[6] (Park et al., 2021), FoodKG[7] (Haussmann et al., 2019), RecipeKG[8]) have been developed in the last decade that facilitates interoperability and increases machine-understandability for the food domain. Creativity in recipes involves multiple techniques that include combining an unusual group of ingredients, replacing ingredients, or attempting alternative cooking methodologies. These techniques are implemented using various models including statistical language models (Antô et al., 2020) and transformers (Gim et al., 2021). In this thesis, we have proposed a Generative approach for evolving Context-Aware Recipes *EvoRecipes*. It is an intelligent cognitive system that uses Genetic Algorithm (GA) (Chambers, 2019, Mitchell, 1998) to generate novel recipes (in RDF format). An initial population of *EvoRecipes* is generated from *RecipeKG* that comprises human-generated recipes in RDF format extracted from recipe websites. Finally, we have used OpenAI GPT API to generate recipe text from RDF format for newly created recipes.

1.1 Motivation

The search for personalized recipes is also becoming more important due to increasing awareness of the impact of food. The requirements are different for

[4] https://foodon.org/
[5] https://hajirajabeen.github.io/EvoRecipesOntology/
[6] https://github.com/lamypark/FlavorGraph
[7] https://foodkg.github.io/
[8] https://hajirajabeen.github.io/RecipeKG

people with allergies, people with diet constraints (e.g. diabetes), people who want to eat healthy, or who want to eat from more sustainable sources. Incorporating these demands dynamically requires computer-assisted intelligent solutions.

The objectives of new recipes and ingredient pairing can range from, specific dietary requirements, cost and health constraints, supporting sustainability, hunger, and to meet nutritional requirements. Besides, new recipes can help the sick to eat better, make a good impact on the environment, and help attain health goals while still being gentle to one's pallet. Sometimes we want to try out new recipes from the list of available ingredients due to time, lock, or cost issues. A cost effective and easily available ingredient substitute can help users in the preparation of dishes. The importance to monitor daily food consumption is twofold due to busy jobs, business routines, and limited exercise facilities. Users can replace the ingredients from recipes that have not have healthy nutrition.

While a multitude of recipe-related ontologies exists but they do not represent the recipes as a process that can be applied to different recipes, furthermore, they do not usually cover the requirements of a specialized recipe search, or assist in building an autonomous system.

There are several recipe datasets published in the last decade by considering different objectives. The main motivation behind building a detailed recipe dataset is to emphasize and highlight the different components involved in the recipes. Our focus is to highlight different interesting patterns like the most frequently occurring ingredient in recipes and its correlation with another frequent ingredient, correlation among ingredients and actions, and different types of actions involved in recipes among their sequences.

Many knowledge graphs have been created in the food domain that focuses on pairing, chemical composition, and nutritional information of ingredients to generate novel recipes. However semantic-aware recipes demand more than just a knowledge-base of ingredients and instructions. Ingredient-ingredient, ingredient-action, and action-action relationships need to be tailored to understand the recipe process and that is focused in $RecipeKG$.

Food is also considered as a computational artifact (Deng et al., 2022) and researchers are exploring the potential of computational creativity to generate novel recipes (Antô et al., 2020, Jabeen et al., 2020, Loughran and O'Neill, 2017). Moreover, with the evolution of the semantic web, machine-understandable recipes have gained more attention in recent years. Several food ontologies (e.g. FoodOn[9] (Dooley et al., 2018), and knowledge graphs (e.g. FlavorGraph[10] (Park et al., 2021), FoodKG[11] (Haussmann et al., 2019), have been developed in the last decade that facilitates interoperability and increases machine-understandability of the food domain. Creativity in recipes involves multiple techniques that include combining an unusual group of ingredients, replacing ingredients, or attempting alternative cooking methodologies. These techniques are implemented using various ML models including transformers (Gim et al., 2021) and evolutionary algorithms (Antô et al., 2020).

This inspires us to develop an intelligent cognitive system that not only makes recipes machine-understandable but also helps in creating novel context-aware recipes that full fills personalized user preferences.

1.2 Problem Statement

The semantic web is about adding semantics to otherwise flat data to make it machine-understandable. However, the enrichment of this semantic information in data for computational creativity has remained under explored. From the user's perspective, the demand for personalized recipes is increasing. This includes specialized food for sick people (e.g, allergic, dieters, or food enthusiasts). Therefore, it is important to explore the semantics underlying recipes to develop computer-assisted recipes. Culinary recipe is a complex process that needs a large scale collection of recipe datasets with a plurality of attributes to understand and model the recipe process. Many knowledge graphs have been created in the food domain that focuses on food-pairing, chemical composition, and nutritional information of ingredients to generate novel recipes. Semantic-aware recipes demand more than just a knowledge base of ingredients and

[9] https://foodon.org/
[10] https://github.com/lamypark/FlavorGraph
[11] https://foodkg.github.io/

instructions. Ingredient-ingredient, ingredient-action, and action-action relationships need to be tailored to understand the recipe process. Generative AI e.g. Large Language Models (LLMs) can be used to generate new recipes. However, LLMs struggle with more complex aspects like recipe semantics and process comprehension. Furthermore, LLMs have limited ability to account for user preferences without hallucinations since they are based on statistical patterns. As a result, these recipes may be invalid.

Evolutionary algorithms inspired by the process of natural selection are optimization algorithms that use stochastic operators to generate new solutions. Evolutionary algorithms can generate a large number of solutions from the set of possible solution space. Moreover, these algorithms have the capability to incorporate user preferences in the fitness function to generate novel recipes that are more aligned with the fitness objective. In this thesis, our aim is to build an intelligent system that comprises supporting modules (like ontology, knowledge graph, and computational creativity) that could help to generate valid semantic-aware recipes to satisfy user preferences.

1.3 Research Questions

This thesis aims to answer the following research questions.

> *RQ1.* How we create a semantically rich context-aware recipe representation that is not only machine-readable but also machine-understandable?

We have presented *RecipeOn*, an Ontology for representing cooking recipes. It follows a modular approach that comprises core-recipe, ingredients, actions, nutritions, and procedure modules. It adds creativity in culinary art by adding semantics to the recipes. It also models the recipe knowledge along with the sequence of actions. *RecipeOn* uses data properties and object properties to define ingredient-ingredient, ingredient-action, and action-action relationships that make a recipe not only machine-readable but also machine-understandable.

> *RQ2.* How we represent the recipe as a process using web semantic techniques (i.e. ontology, knowledge graphs)?

RecipeOn ontology successfully maps any given recipe as a process and is capable to answer the competency questions designed for intended users (sick persons, diet-conscious people, and food enthusiasts). *RecipeOn* makes it easy for the users to follow the recipe steps using ingredient-actions, and action-action relations. *RecipeOn* helps in customized recipe generation and is designed following NeOn methodology for ontology development while using the classes and properties from existing ontologies (i.e, owl, Seq, schema, and Qudt).

> *RQ3.* Is there any recipe dataset with a plurality of attributes that can be used to create a recipe knowledge graph?

Due to the complexity of the cooking process, learning of culinary recipes requires a large scale collection of recipes (like Recipe1M+[1] (Marin et al., 2019)). In recent years, the volume of the recipe datasets has increased from a few thousands (Bossard et al., 2014, Chen et al., 2009, Jain, 2020) to million plus items (Harashima et al., 2016, 2017, Salvador et al., 2017). However, less attention has been paid to rich recipe features (i.e. relevant data and metadata) extraction. Understanding culinary recipes demands rich metadata. This may include descriptive metadata (like published date, author, cooking time, title, etc) or contextual metadata (utensils used during an action). We have selected top 500 (previously based on alexa[12] rank, now similarweb[13] score) recipe websites for data extraction. Data is extracted using Visual Web Ripper (VWR) [14]. Extracted data contains four main entities that contain 0.8 Million recipes with 50+ attributes.

> *RQ4.* How computational techniques can be used to generate context-aware novel recipes that are also capable to accommodate user preferences?

[12] https://www.alexa.com/
[13] https://www.similarweb.com/
[14] http://www.visualwebripper.com, Accessed Oct 02, 2021

In this thesis, we have proposed a *EvoRecipes* framework that uses both evolutionary algorithm and generative AI to create custom context-aware recipes. Evolutionary algorithms are based on the phenomena of survival of the fittest and calculate the fitness of an individual using a fitness function. Therefore, fitness functions are tuned to accommodate user choices and preferences. Moreover, recipes have been encoded using *RecipeOn* ontology which makes them not only machine-readable but also machine-understandable.

> *RQ5.* How can we evaluate the quality of machine-generated novel recipes?

To evaluate the alignment of *EvoRecipe* generated recipes with multiple intended objectives, we propose a fitness function that can incorporate multiple objectives. Additionally, to evaluate the quality of the *EvoRecipe* generated recipes while considering the subjective nature of recipes, we conducted a survey using multi-dimensional metrics (i.e., contextual, procedural, novelty) to assess the effectiveness of *EvoRecipes*.

1.4 Thesis Contribution

This thesis contributed to developing a recipe ontology *RecipeOn*, a recipe knowledge graph *RecipeKG*, and a recipe evolution framework *EvoRecipe*.

- *RecipeOn*[5] *Ontology:* We have developed *RecipeOn* ontology that not only helps to increase machine understandability but also encodes important information about the recipe. *RecipeOn* ontology guides the user in preparing recipes as a systematic process. It provides detailed information related to ingredients, actions, nutrition, and the sequence of actions and ingredients in each procedure. *RecipeOn* helps evolve new recipes and personalize the existing recipes for different types of user preferences using ingredient-action relations and action-action relations.
- *0.8M-Recipes*[15] We have compiled 0.8 million recipes with a plurality (50+) of attributes. It contains 11 million + ingredients, 6 million+ instructions,

[15] https://github.com/HajiraJabeen/RecipeKG/tree/main/0.8M-Recipes

and 0.5 million nutrition details. We have extracted data from the top 500+ recipe websites and applied, data consolidation, data pre-processing, and data transformation on extracted recipes.

It contains basic recipe details, a list of ingredients, nutritional information, and instruction statements.

- *RecipeKG* [16]*:* A Recipe Knowledge Graph *RecipeKG* is built using the 0.8M-Recipes. *RecipeKG* uses the concepts and relationships as specified in *RecipeOn* ontology. It represents the recipe as a process and defines the ingredient-action, action-action, and ingredient-action relationships. *RecipeKG* is based on a huge collection of 52.96 million instances and 209 million facts.

- *EvoRecipes* [17]*:* An intelligent cognitive framework that evolves and generates new recipes using Genetic Algorithms. Furthermore, *EvoRecipes* proposes qualitative metrics to evaluate the subjective parameters of novel recipes. It also maps the recipe RDF to recipe text using OpenAI GPT.

1.4.1 List of Publications

This thesis has one accepted publication, while two are submitted details are as under. These publications answered the research questions listed in the above section.

Muhammad Saad Razzaq, Fahad Maqbool, Hajira Jabeen, RecipeOn: An Ontology to Represent Recipe as a Process submitted in Expert System with Applications, Elsevier, 2022. Included in Chapter 4.

Muhammad Saad Razzaq, Fahad Maqbool, Hajira Jabeen, *RecipeKG*: Knowledge Graph to Manifest Culinary Recipes, to be submitted in Knowledge-Based Systems, Included in Chapter 5.

Muhammad Saad Razzaq, Fahad Maqbool, Muhammad Ilyas, Hajira Jabeen, A Generative Approach for Evolving Context-Aware Recipes, published in IEEE

[16] https://github.com/HajiraJabeen/RecipeKG

[17] https://github.com/HajiraJabeen/EvoRecipesOntology/blob/main/EvoRecipes/EvoRecipes.py

Access, vol. 11, pp. 74148-74164, 2023, doi: 10.1109/ACCESS.2023.3296144. Included in Chapter 6.

1.5 Thesis Structure

This thesis consists of seven chapters. Chapter 1 includes the introduction, motivation, problem statement, research questions, thesis contribution, and publication details. In chapter 2 we have briefly covered the concepts that are used in the thesis. In chapter 3 we have highlighted the related work relevant to food ontologies, recipe datasets, food knowledge graphs, and recipe evolution. In chapter 4 we have covered our proposed ontology *RecipeOn* that represents recipe as a process. In chapter 5 we have explained the proposed knowledge graph *RecipeKG* and 0.8 million recipe dataset. Recipe evolution framework *EvoRecipes* is presented in chapter 6. In chapter 7 we have summarized the thesis and discussed the future direction.

Chapter 2

Preliminaries

In this chapter, we have briefly explained the concepts that are involved in this thesis.

2.1 Food & Culinary Recipes

Food is a necessity for humans, and it is being consumed in different ways across the globe from the start of the universe. The human body requires an adequate amount of air, food, and water for living and performing daily essentials. Food not only provides nutrition, vitamins, minerals, and salt to the human body but it is also a source of energy for the human body. Food can be prepared using recipes that comprise a set of items and actions. Recipes have their unique characteristics (such as taste & aroma) and many distinguishing factors (such as nutrition, cooking time, serving size, and recipe yield). Recipes have an ancient history from being recorded on cuneiform tablets to modern television and internet-based recipes. Television channels (like Cooking Channel[1], Food Network[2]) , television programs(like Top Chef[3]), and recipe websites (like All Recipes[4], Epicuruous[5], Bon Appetit[6]) are still a source of many modern recipes.

The taste of a recipe is based on multiple factors (i.e. cooking method, style, regional culture, actions, availability of ingredients, and chemical composition of ingredients). Taste is generally considered the most significant criterion to evaluate a cooking recipe. It is not only affected by the combination of ingredients (i.e. and their chemical composition) but also by the actions that you

[1] https://www.cookingchanneltv.com/
[2] https://www.foodnetwork.com/
[3] https://www.bravotv.com/top-chef
[4] https://www.allrecipes.com/
[5] https://www.epicurious.com/
[6] https://www.bonappetit.com/

perform on these ingredients. A sequence of actions and the duration of each action also play a key role in developing the taste of a recipe (Ahn et al., 2011).

Food pairing is an important factor that is considered by food scientists while preparing novel recipes. One should have to take care of the chemical composition, flavor, color, and texture of the ingredients that are combined in a recipe. So for novel recipe creation, ingredient chemistry along with the preparatory actions that modify the ingredients have a vital role in the flavor and creativity of recipe (Antô et al., 2020), and (Park et al., 2021).

2.2 Semantic Representation & Technologies

The primary goal of the world wide web was to manage data and information. Initially, there was not a proper and defined structure available for storing, organizing, and updating the data and information over the web. The main focus of web-based systems was to help humans to get organized information over the internet. However, these web-based applications do not understand the meaning of the information. Existing web-based applications have difficulties in information retrieval, information extraction, maintenance, and personalization. To overcome these difficulties web semantics was introduced. Web semantics maps and encode the information in a Resource Description Framework (RDF) subject-predicate-object (triplet) format that helps them in making it machine-understandable. It also adds metadata information with data that helps in making them machine-understandable. It helps in getting more precise results by enabling the relationships and meaning among desired concepts and entities gathered from various similar resources. The semantic web enables interoperability among different domains by using standardized ontologies and languages.

2.2.1 Resource Document Framework (RDF)

RDF is used to represent structured information. Its main focus is to represent relationships among objects and present them in the form of a graph. RDF uses Uniform Resource Identifier (URI) to identify the objects uniquely. Listing 2.1

shows the rdf triplets where a recipe node is connected with different ingredient and action nodes.

LISTING 2.1: RDF Schema of a Recipe

```
Recipe          hasIngredient           Ingredient .\
Recipe          hasProcedure            Procedure .
Ingredient      hasAction               Action .
Action          directlyPrecedes        Action .
Procedure       hasProcedureIngredient  Ingredient  .
Procedure       hasProcedureAction      Action  .
```

2.2.2 Ontologies

Ontologies are used to represent relationships among concepts. They help in creating a data model that is based on concepts of any domain and identifies the relationship among these concepts. Ontologies are based on concepts (classes), class hierarchies, relations, relation hierarchy, and instances (individuals). Ontologies help in describing and formulating domain knowledge in terms of classes, attributes, relationships, rules, and properties (Taye, 2010). Ontologies help in semantically connecting concepts and defining relationships among them. Ontologies also facilitate structuring the data, which makes it understandable by humans and machines. It also solves the problem of interoperability among domains. There are several tools (Protege, OntoEdit, WebOnto, Hozo, and OWLGrEd) and languages (Resource Description Framework (RDF), SPARQL, Web Ontology Language (OWL), turtle, and XML) are available for ontology development (Mizoguchi, 2004, Slimani, 2015). Ontologies are actively used in Natural Language Processing tasks, Artificial Intelligence, Multi-Agents, Knowledge Engineering Management, and Web Semantics (Slimani, 2015).

2.2.3 SPARQL

SPARQL Protocol and RDF Query Language (SPARQL) is a query language that is used to query from a data source that is mapped in RDF format or from databases. It can be used and executed on the database that is viewed in RDF. SPARQL is also an HTTP-based transport protocol. It accesses multiple data sources and enables queries on nonuniform data over distributed resources. It helps in designing advanced queries across distributed data sources. For ease of use, SPARQL uses the same keywords for different query operations as in other query languages. Like select, group by, order by, having, limit, and distinct are used in SPARQL, SQL, and several other languages. Moreover, SPARQL also used query keywords that are specific to RDF. SPARQL[7] is following all the recommendations suggested by W3C.

2.2.4 Knowledge Graph

Real-world knowledge is situational (meaning varies with scenarios and application areas), associative (links between concepts create a more clear understanding), and evolving (new additions to knowledge may change meanings). Semantics is the key that separates data from knowledge. A knowledge graph is a graph-structured representation that makes your data machine-understandable. A knowledge graph (KG) is a graph-structured knowledge base that interlinks the entities through well-defined semantic descriptions. KG is a combination of nodes, edges, and labels. Nodes are entities that are combined with edges (i.e. show the relationship among entities) and are defined by labels. KG is used to organize, manage and integrate the information from various structured and unstructured resources.

Ontology is the heart of a knowledge graph that specifies the semantics by defining the concepts and relationships to interlink those concepts. Figure 2.1 shows the knowledge graph against the fluffy microwave scrambled eggs recipe. it is obvious from the figures that the knowledge graph is semantically rich and more helpful in understanding the recipe process.

Listing 2.2 represents triplet i.e. subject, predicate, and object relationship in RDF graph as shown in figure 2.1.

LISTING 2.2: RDF Triplet of Fluffy Microwave Scrambled Eggs Recipe

```
FluffyMicrowaveScrambledEggs          hasIngredient          Egg .
FluffyMicrowaveScrambledEggs          hasIngredient          Milk .
FluffyMicrowaveScrambledEggs          hasIngredient          Salt .
```

[7] http://www.w3.org/TR/rdf-sparql-query/

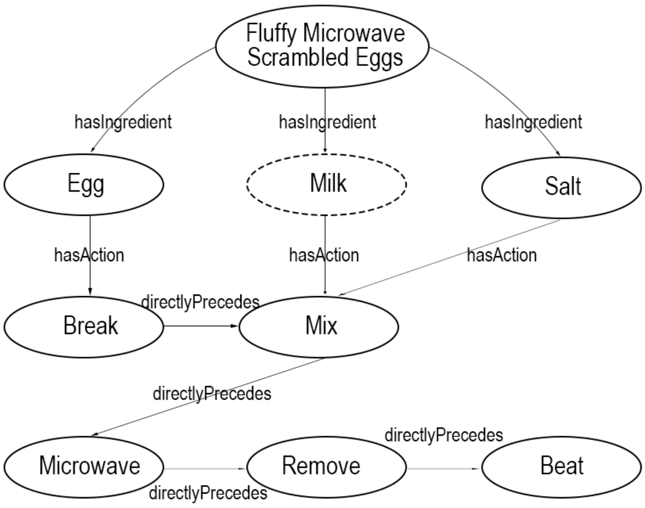

Figure 2.1: An RDF Graph of Fluffy Microwave Scrambled Eggs Recipe

```
Egg         hasAction          Break .
Milk        hasAction          Mix .
Salt        hasAction          Mix .
Break       directlyPrecedes   Mix .
Mix         directlyPrecedes   Microwave .
Microwave   directlyPrecedes   Remove .
Remove      directlyPrecedes   Beat .
```

2.3 Computational Creativity & Evolutionary Algorithms

Creativity in culinary recipes has gained more importance in recent years due to the increasing interest of people in food culture and memorable dining experiences. Demand for specialized recipes from sick persons, diet-conscious people, and food enthusiasts also complements culinary creativity. Moreover, the rise of the social web has made it easier for chefs to share their newly generated recipes (based on their experiences, intuition, & imagination) and get recognition for their culinary creativity. However, human creativity in culinary recipes has a few limitations: *(i)* Error - Humans may produce errors while creating a new recipe or may misjudge the outcome of a recipe; *(ii)* Biasness - Chefs may have biasness for certain ingredients based on their availability in a certain region or certain recipes that are well received by the customers. Similarly, chefs may have preferences for certain cooking methods based on their skills limitations, or experiences. These factors play a significant role in limiting creativity; *(iii)* Speed - Human chefs need more time and effort to generate novel recipes than computational alternatives. On the other hand, computational creativity in culinary recipes overcomes human errors, is not affected by biases, and can generate recipes at a much faster pace than humans.

Evolutionary algorithms inspired by the process of natural selection are optimization algorithms that use stochastic operators to generate new solutions. Evolutionary algorithms can generate a large number of solutions from the set of possible solution space. Moreover, these algorithms can incorporate user preferences in fitness function to generate novel recipes that are more aligned with the fitness objective.

In the following subsection, we have discuss creativity in culinary recipes and the genetic algorithm (an evolutionary algorithm) in more detail.

2.3.1 Computational Creativity in Culinary Recipes

Creativity is a complex concept that is hard to define and difficult to evaluate. It not only refers to creating something new but may also relate to solving old problems in a new way. Alongside the creation of new ideas, creativity also focuses on the application of those ideas. Creativity adds value to the solutions by improving their quality and acts like a force that compels evolution and change. Scientists modeled the concept of creativity to computer science using different techniques and algorithms to simulate human creativity and named it computational creativity (Ławrynowicz, 2020). Computational creativity is based on concepts from psychology, arts, and philosophy. Computational creativity is used to generate new ideas: By combining existing solutions or their parts to create new solutions that did not exist; By transforming the solutions to generate novel solutions that were difficult to be created using the traditional creativity process; By exploring high dimensional conceptual spaces that are difficult to be explored using traditional creativity process. Computational creativity doesn't assure that computers are creative but demonstrates that computers can model and simulate human creativity to generate novel ideas/solutions. Linguistic creativity, music creativity, visual art creativity, and creative problem-solving are the few success stories of computational creativity (Loughran and O'Neill, 2017).

Cooking is an interesting case study to explore computational creativity. Cooking is referred to both as an art and a science. It's an art as there is too much subjectivity involved in terms of taste, color, texture, aroma, and style. It's a science as it involves chemical components present in ingredients and many chemical processes, to transform ingredients into edible food. Creativity in culinary recipes can be explored in multiple contexts: (1) Changing ingredients or a combination of ingredients to create a novel recipe; (2) Exploring multi-dimensional ingredients search space to find a potential ingredient combination that could generate a novel recipe. Humans generally do this using intuition as the human mind is not capable to explore large search space; Applying new actions on existing ingredients that may change the aroma; Replacing the sub-process (that involves both ingredients and actions) of a recipe with an alternate potential sub-process.

2.3.2 Genetic Algorithms

Evolutionary Algorithms (EAs) are widely used in Machine Learning to solve real-life problems. EAs have multiple techniques i.e. (Genetic Algorithms, Genetic Programming, Evolutionary Programming) that are used for solving simple to complex optimization problems and designing a new solution to a problem. EAs have a set of operations that are performed in the generation to evolve the solution. It involves initializing the population, selection, crossover, mutation, and fitness evaluation of the solution. Genetic Algorithm (GA) is one of the popular EAs and is widely used for solving different problems due to its simple structure (Mitchell, 1998) and wide adaptability for solving different real life problems. GA used (Initialize, selection, crossover, mutation, solution evaluation) across the generations to solve the problem. In the initialization phase, the initial population is generated, while in the selection operation, individuals are selected for the next generation based on some selection criteria. In crossover, GA combines the traits of current solutions to create the new solutions, while in mutation values of individual solution is changed to generate the new solutions. GA used a directed randomized approach that helps in finding an optimal solution by avoiding the local optima. GA is empowered with the capability of exploring the search space and exploiting the solutions with crossover and mutation. The quality of the solution is evaluated by the fitness function that is designed according to the problem domain. The termination criteria of GA is reached by satisfying the convergence criteria or by reaching the maximum number of generations. Algorithm 1 states the main steps involved in Genetic Algorithm.

Algorithm 1 Genetic Algorithm

$P \leftarrow CreateInitialPopulation$

while Termination Criteria not reached **do**
  $P` \leftarrow SelectionofIndividuals$(P)
  $P` \leftarrow ApplyCrossover$(P`)
  $P` \leftarrow ApplyMutation$(P`)
  $P` \leftarrow Survival - Selection(PUP`)$
end while

Chapter 3

Related Work

Culinary cooking is a complex task dependent on many factors (selection of ingredients, proportions of ingredients, substitution of ingredients, actions on ingredients, sequence of actions, and relationship between actions). Food experts say that the simplest recipes (e.g. boiled rice, baked chicken, french fries) are often difficult to cook as there is less number of choices to hide behind fancy items (i.e. sauces, sprinkles, garnish). This further increases the significance of the above-mentioned factors. In this chapter, we have discussed the literature relevant to recipe ontologies, recipe datasets, food knowledge graphs, and recipe evolution strategies. We have tried to highlight the research gap and need for the development of a recipe ontology *RecipeOn*, recipe knowledge graph *RecipekG*. We have also discussed the shortcomings in existing recipe evolution strategies and the need for the development of the *EvoRecipes* framework.

3.1 Food Ontologies

Many food-related ontologies have been developed in the past. FOODS (Food-Oriented Ontology-Driven System) (Snae and Bruckner, 2008) ontology was developed to act as a counseling system for food or menu planning in a restaurant, clinic/hospital, or at home. Recipe data sets are used in developing food ontologies, a similar effort is made by FoodOn (Dooley et al., 2018) a food Ontology that comprises ingredients, cooking instructions, packaging, and preservation details. FoodOn also covered the chemical composition of food due to its interoperability with agricultural, species, agronomic and anatomical ontologies. FoodOn helps in food traceability, quality control, and recipe analysis. It also has a focus on food safety, food security, and agricultural and animal husbandry practices linked to food production. FoodOn didn't focus on the steps, that helped in preparing the Recipes. BBC Food Ontology (BBC, 2014) is created for publishing data about recipes, menus, seasonal influences, courses, and related occasions. FoodWiki (Foodwiki, 2014) was developed to help individuals in risk groups avoid foods by helping them automate the decision-making process of what and how much to eat, as well as what to avoid, based on the nutrient profile and especially the presence or absence of potentially dangerous additives (such as color, preservers, sweeteners) added to many processed food that is contributing to severe diseases like cancer, diabetes, and cardiac issues. Food Product Ontology (Kolchin and Zamula, 2020) describes food product metadata that is used by manufacturers, retailers, and food-related institutions. WhatToMake ontology (Haussmann et al., 2019) was developed using the FoodKG knowledge graph. They have detail about the components of a Recipe including cook time, meal type, and ingredients. WhatToMake ontology is also used for ingredient substitution based on allergies. Bionutrition Ontology (Musen et al., 2012a) was developed to suggest diet plans based on specified nutrient details. ISO-Food ontology (Eftimov et al., 2018) is based on food nutrient details and connected to other food ontologies that contained isotopic measurements related to food items.

It is important to note that none of these existing ontologies cover the recipe as a process and handles the necessary aspects desired for efficient searching, or preparing food. Considering this gap, in this thesis we have proposed *RecipeOn* ontology that not only facilitates the semantic comprehension of the recipe process but also makes this process machine-understandable, assisting machines to understand and create novel recipes. There are many approaches available for ontology development (such as top-down, bottom-up, and middle-out). We have followed a bottom-up approach considering its simplicity and requirements while developing *RecipeOn* ontology. The bottom-up approach starts with simple competency questions while generating more complex competency questions. We have also consulted a few question-answering articles in the cooking domain (Manna et al., 2021) (Rahman Khilji et al., 2020) (Manna et al., 2016) for developing these competency questions. We have followed the NeOn methodology for ontology Engineering (Suárez-Figueroa et al., 2012) in RecipeOn development. NeOn provides a flexible approach to ontology development and presents different scenarios for ontology development. The neOn methodology focuses on reusability, collaborative work, re-engineering of knowledge resources, reusing ontology & nonontological resources, restructuring ontological resources, localizing the ontological resources, and merging the ontological resources. There are totally eight tasks in the requirements phase of ontology engineering (Pérez et al., 2008). These tasks and their details are shown in chapter 4. All of the above Ontologies are limited concerning the detailed information required to represent the Recipe as a process. In this thesis we have presented *RecipeOn* ontology in chapter 4, which models a diverse set of recipes from different cuisines, taking into account the ingredients, their types, the actions applied on ingredients, the nutrition information, and the sequence of the cooking process. We have also presented the use case of *RecipeOn* by generating a novel recipe for a sick person and suggested a Recipe with alternative ingredients, actions, and procedures.

3.2 Recipe Datasets and Food Knowledge Graphs

Due to the growing trend of information technology gadgets in every facet and domain, recipe knowledge and information is also a single click away. Audience search and look for a variety of information about recipes (like preparatory steps, ingredients, cooking time, temperature, cooking instructions, quantity). Therefore, rich recipe metadata and information is useful for culinary recipe seekers. Table 3.1 lists a few recipe datasets with recipe count, number of attributes, data format, and nutritional information. Yummly [1] contains above 66000 recipes with attributes recipe name, image, cuisine, course attributes, and ingredients information. However, Recipe directions, nutritional information, author information, and user interactions are not provided in Yummly. The dataset is publicly available in JSON format. Recipe1M+ (Salvador et al., 2017) (Marin et al., 2019) contains over one million recipes along with 13 million food images collected from two dozen recipe websites and from various image resources. However, the dataset has limited attributes (i.e. title, ingredients, instruction, unit, quantity, and recipe images). Moreover, they have

[1]http://yummly.com

used embeddings to match the recipe ingredients and instructions with images. Recipe1M+ dataset is used by RecipeNLG (Bień et al., 2020) in creating their dataset along with the help of a few other websites. Above two million different recipes are collected with attributes including name, ingredients, instructions, and source URL. However Nutritional information, recipe image, author information, and user interactions are not part of the dataset. Another dataset Food.com (Majumder et al., 2019) contains 230K+ recipes along with 1M+ user interactions. Recipe data contains recipe-name, ingredients, instructions, time, and nutritional information whereas interaction data contains user-id, recipe-id, date, rating, and user review. The recipe image and author information is not part of the dataset.

TABLE 3.1: Recipe Datasets with Number of Records and Attributes

SN.	**Dataset Name**	**Recipe Count**	**Attributes**	**Nutritional Information**	**Data Format**
1	Yummly [2]	66,61a 5	6	No	JSON
2	Recipe1M+ (Salvador et al., 2017) (Marin et al., 2019)	1 Million +	5	No	NPY
3	Recipe NLG (Bień et al., 2020)	2,231,142	5	No	CSV
4	Food.com (Majumder et al., 2019)	180K+	12	Yes	CSV
5	Recipe DB (Batra et al., 2020)	118,171	15	Yes	–
6	Harashima's text corpus (Harashima et al., 2016)	1.7 million	6	No	MySQL
7	Epicurious [3]	over 20K	4	Yes	CSV
8	Food Recipe Dataset [4]	82K	9	No	CSV
9	Recipe Ingredients Dataset [5]	50,000	3	No	JSON
10	Recipe Dataset by E-TNC (Ahn et al., 2011)	56,458	3	No	RDF

RecipeDB (Batra et al., 2020) is a structured dataset that contains 118171 recipes with 23548 ingredients. Collected recipes are classified into cuisines based on 26 geo-cultural regions, ingredients are categorized into 29 categories. Recipes are also labeled with 5 dietary styles and annotated for 268 cooking processes. The food recipe dataset[6] contains over 82000 salad recipes with recipe names, ingredients, instructions, image, prep time, cuisine, servings, and keywords. Epicurious [7] dataset contains 20,000+ recipes with recipe title,

[6] http://kaggle.com/snehallokesh31096/recipe
[7] https://epicurious.com

ingredients, instructions, nutritional information, recipe rating, and recipe category. This dataset is publicly available in CSV, JSON, and Py formats. In Harashima's text corpus (Harashima et al., 2016), there are 1.7 million recipes and 36000 meals. This corpus includes recipe title, description, ingredients, steps, advice, and history. This dataset is limited to Japanese recipes only. Recipe-Ingredients[8] dataset is published by Yummly to recognize recipe cuisine based on the ingredients list. The dataset contains recipe id, cuisine, and ingredients list. Moreover, the dataset contains approximately 50000 recipes and is publicly available in JSON format. Recipe dataset published by E-TNC (Ahn et al., 2011) is based on RDF format and contains recipe cuisines and a list of ingredients. Recipe count of the dataset is 56,458 that are divided into 11 cuisines. 6000+ Indian Food Recipes dataset [9] contains recipe name, ingredients, instructions, timing information, course, cuisine, and serving information.

Due to the complexity of the cooking process, learning of culinary recipes requires a large scale collection of recipes (like Recipe1M+[1] (Marin et al., 2019)). In recent years, the volume of the recipe datasets has increased from a few thousands (Bossard et al., 2014, Chen et al., 2009, Jain, 2020) to million plus items (Harashima et al., 2016, 2017, Salvador et al., 2017). However, less attention has been paid to rich recipe features (i.e. relevant data and metadata) extraction. Understanding culinary recipes demands rich metadata. This may include descriptive metadata (like published date, author, cooking time, title, etc) or contextual metadata (like utensils).

Recipe datasets are helpful in recipe representation (Li and Zaki, 2020), recipe generation (Antô et al., 2020), recipe ingredient replacement (Nadee and Unankard, 2021), (Pan et al., 2020), recipe recommendation system (Khilji et al., 2021), Food recommendation based on knowledge Graph (Park et al., 2021) and food pairing recommendations (Park et al., 2021). Moreover, recipe datasets are widely being used in deep learning algorithms and computer vision techniques (Bień et al., 2020). Not only these datasets are large scaled but also have high memory requirements. Researchers created customized datasets based on the applications in different domains (such as recipe recommendation system (Lei

[8] http://kaggle.com/kaggle/recipe-ingredients-dataset
[9] http://data.mendeley.com/datasets/xsphgmmh7b/1

et al., 2021), healthier recipe recommendation (Elsweiler et al., 2017), image to recipe retrieval (Chen et al., 2018), recipe matching with images (Chen et al., 2017), and ingredients recognition (Bolaños et al., 2017)).

Knowledge Graph (KG) is a knowledge base that uses graph structure to store data. It's a network of related and real-world entities connected through some predicate/relationship. Knowledge graphs are not only machine-readable but also machine-understandable.

Food recommendations based on knowledge graph are popular among scientists and researchers in the food industry. FlavorGraph is a large scale graph proposed for food pairing recommendations and food relation predictions (Park et al., 2021). It is a collection of food ingredients and their chemical compounds. The nodes consist of flavor compounds, drug compounds, and food ingredients while the edges comprise (ingredient-ingredinet, ingredient-flavor compound, and ingredient-drug compound) relations. Food pairing recommendation is based on the chemical similarity of food ingredients. FlavorGraph is based on various datasets related to food recipes. A major shortcoming of FlavorGraph is that it requires more food relations for better pairing and predictions. Moreover, it lacks a scientific method for the evaluation of food pairing and prediction. A recipe recommendation system through a recipe knowledge graph (RegKG) was proposed by Lei et.al (Lei et al., 2021). RegKG is constructed according to user needs, requirements, user hobbies, and recipe characteristics. RegKG recommendations performance was improved with the help of images and videos. FoodKG (Haussmann et al., 2019) suggests recipes based on consumer choice. FoodKG is based on recipes along with ingredients, nutritional details, and Food Ontology. Different competency questions related to ingredient entity are answered by FoodKG in SPARQL and natural language. However, FoodKG does not cover the recipe as a process and does not specify the actions and their sequence. As existing recipe datasets are limited in information and very few recipe datasets have been mapped to knowledge graphs. As underlying datasets do not provide a plurality of attributes therefore knowledge graphs created on top of these datasets bear a limited knowledge base.

In this thesis, our focus is to develop a large scale recipe dataset with a rich set of attributes. This dataset would contain recipes from the top (based on internet traffic) recipe websites. Moreover, we will build an information rich Recipe KG $RecipeKG$. Attributes naming convention would be standardized and followed as in schema.org to improve the interoperability and reusability of $RecipeKG$.

3.3 Recipe Evolution

Exploring new and unique recipes is always in demand by food enthusiasts, health-conscious individuals, patients, and chefs. Novelty in recipes is being pursued with the help of computer-aided and artificial intelligence-based techniques. Computational creativity, a sub-field of artificial intelligence, has also been explored in the culinary domain. Various aspects of culinary science such as computational gastronomy (Shukla and Ailawadi, 2019), flavor gastronomy (Varshney et al., 2019), flavor pairing (Spence, 2022), food perception (Schifferstein et al., 2022), flavor perception (Shao et al., 2014), food pairing (Al-Razgan et al., 2021, Gim et al., 2022), knowledge networks for robotic cooking (Sakib et al., 2022), and sustainable food systems (Camarena, 2020) have been investigated using computational creativity. The perception of food and flavor is influenced by several factors including the chemical relationships among ingredients, color, shape, temperature, and texture. In the culinary domain, computational creativity involves pairing and mixing ingredients in novel ways (Jimenez-Mavillard and Suarez, 2022, Lawo et al., 2020, Morris et al., 2012, Pinel and Varshney, 2014, Shirai et al., 2021, Tuwani et al., 2019). Identifying associations among ingredients in different recipes, analyzing the co-occurrence of flavor compounds and colors (Varshney et al., 2016), identifying frequent patterns among cuisines (Sharma et al., 2020), utilizing cognitive informatics (Varshney et al., 2013), generating cooking actions from recipes (Venkataramanan et al., 2023), suggesting cross-cultural food preferences (Zhang et al., 2023), and diet recommendation system (Reddy et al., 2023) has recently been explored in the culinary domain. The internet is abundant with information, making it challenging to integrate all pertinent information with provenance for optimal food choice recommendations. Traditionally, the food pairing choice of food experts depends on their experiences.

However, Artificial Intelligence (AI)-based recommender systems have recently emerged as an alternative for providing food pairing recommendations. FlavorGraph (Park et al., 2021) is one such system, a large-scale food-chemical graph that utilizes AI to predict relationships between food and chemical compounds. Its nodes contain information on either ingredients or chemical compounds, while the edges represent relationships among ingredients and chemical compounds. Another food recommendation knowledge graph FoodKG (Haussmann et al., 2019) provides personalized food recommendations based on user preferences (such as ease of preparation, convenience, spiciness, crispiness, and other generic requirements). However, FoodKG did not recommend and explore relationships related to actions and procedures involved in the recipe.

A recommendation system (Gim et al., 2021) for ingredient and recipe selection has been developed that utilizes cooking tags and ingredients as inputs to suggest related options. However, due to limitations in the system's training on cooking-related knowledge, it sometimes recommends inconsistent recipe choices given a particular set of ingredients. In another effort toward recipe evolution (i.e. SmartChef (Draschner et al., 2019)), a genetic programming approach has been employed to evolve the recipes encoded in a tree structure. The authors have tested their approach on a 128-recipe dataset. They didn't consider the semantics of recipes while generating new recipes. Moreover, they have designed a semi-automated fitness function. EvoChef (Jabeen et al., 2019) is an evolutionary approach to recipe generation that relies on recombining ingredients and instructions to generate new recipes, but this system has an initial population of 08 recipes dataset consisting solely of potatoes. Additionally, it lacks critical information such as nutritional information, flavor, and ingredient pairing details. They validated recipes just through human feedback. In another effort, Jabeen et. al. proposed AutoChef which recommends recipes using genetic programming (Jabeen et al., 2020). They have used a dataset of 75 recipes. However, AutoChef has limited ingredient and action replacement options. Moreover, AutoChef did not focus on the sequence of actions while performing the mutation on the action node.

W. Anto et al. proposed an automatic recipe generation using genetic programming along with a language model used for the decomposition of recipe and then recomposing them to generate new recipes (Antô et al., 2020). Authors worked on a limited set of recipes and while recombination of recipes they used a limited list of ingredients and preparatory actions. They didn't ensure the proper classification of ingredients, actions, and instructions for generating new recipes. A different recipe recommendation system based on user questions was proposed by khilji et al. (Khilji et al., 2021). They have developed a question classification model through which user question is assigned a label. This class label is used by the recommendation engine for recipe recommendation. They have trained the model on a limited recipe dataset and the system is unable to classify the user questions into more specific categories. Therefore recipe recommendation is more general rather than specific to user questions. Novelty in recipes is also generated by updating actions, ingredients, and sequences of actions. A similar concept recommending alternative ingredients for Thai cuisine with suitable ingredients was proposed by (Nadee and Unankard, 2021) to generate new recipes. Another alternative ingredient replacement was proposed by y. Pan et al. (Pan et al., 2020). They replaced ingredients based on resemblance estimation. All of these techniques used computational creativity to generate new recipes, including food pairing, flavor pairing, computational gastronomy, and ingredient substitutions.

Existing approaches use small-sized recipe datasets with a limited number of attributes. These dataset comprises of recipes represented in a similar or limited format extracted from a few recipe websites that restrict the model's capability to improve the machine-understandability of a recipe. These approaches attempted ingredient substitution based on a limited ingredient set and similarly substituted actions based on a limited number of actions to generate new recipes. Also, these approaches lack in expressing the sequence of actions, the relationship between ingredients, and the relationship between ingredients and actions. This can have a significant impact on the taste, aroma, and visual presentation of food items. Moreover, existing techniques focus on the improvement of machine-readability while compromising the machine-understandability of data. Furthermore, many existing approaches also lack in quantifying the semantic validity of novel recipes. Finally, the machine-generated novel recipes are difficult to understand by humans as these are in a machine-readable format. In this thesis, we have covered all of these research gaps and generated novel recipes using *EvoRecipes* that are semantically valid, and available in a human readable format. Moreover, *EvoRecipes* generated recipes are machine-understandable and are based on *RecipeOn* ontology that has rich recipe knowledge and provides the schema for ingredient-action and action-action relationships.

Chapter 4

RecipeOn: An Ontology to Represent Recipe as a Process

The semantic web is about adding semantics to otherwise flat data to make it machine-understandable. However, the enrichment of this semantic information in data for computational creativity has remained under-explored. This chapter covers creativity in culinary arts by adding semantics to the recipes, not only making recipes machine-understandable but also assisting machines in generating new recipes. From the user's perspective, the demand for personalized recipes is increasing. This includes specialized food for the sick (e.g., allergies), diet-conscious, or food enthusiasts. Therefore, exploring the semantics underlying recipes to develop computer-assisted recipes is important.

Creativity is a complex concept that is hard to define and difficult to evaluate. It not only refers to creating something new but may also relate to solving old problems in a new way. Alongside the creation of new ideas, creativity also focuses on the application of those ideas. Creativity adds value to the solutions by improving their quality and acts like a force that compels evolution and change. Scientists modeled the concept of creativity to computer science using different techniques and algorithms to simulate human creativity and named it as computational creativity. Computational creativity is based on concepts from psychology, arts, and philosophy. Computational creativity is used to generate new ideas: By combining existing solutions or their parts to create new solutions that did not exist; By transforming the solutions to generate novel solutions that were difficult to be created using the traditional creativity process; By exploring high dimensional conceptual spaces that are difficult to be explored using traditional creativity process. Computational creativity doesn't assures that computers are creative but demonstrates that computers can model and simulate human creativity to generate novel ideas/solutions. Linguistic creativity, music creativity, visual art creativity, and creative problem-solving are the few success stories of computational creativity.

In this chapter, we have covered the following research questions.

> *RQ1.* Can we create a semantically rich context-aware recipe representation that is not only machine-readable but also machine-understandable?

> *RQ2.* Can we represent the recipe as a process using web semantic techniques (i.e. ontology, knowledge graphs)?

This chapter has the following contribution.

- *RecipeOn* adds creativity to culinary art by adding semantics to the recipes.
- *RecipeOn* can help to generate novel recipes using alternative ingredients and actions.
- *RecipeOn* models the recipe knowledge along with the sequence of actions.
- *RecipeOn* encodes the information crucial for personalized recipe generation. This includes ingredient-action relations and action-action relations.

This chapter is based on the following publication.

> ***Muhammad Saad Razzaq****, Fahad Maqbool, Hajira Jabeen, "RecipeOn: An Ontology to Represent Recipe as a Process", submitted in Expert System with Applications, Elsevier, 2023.*

This chapter is structured as follows. Section 4.1 explained the development of *RecipeOn* ontology and its requirements. In section 4.2 explained the ontology overview, core recipe, ingredients, actions, nutrition, procedures, and property restrictions. In section 4.3 ontology evaluation and its quality parameters are discussed. In section 4.4 *RecipeOn* example is presented. The usecase of *RecipeOn* ontology is presented in section 4.5 and the summary is presented in section 4.6.

4.1 Development of RecipeOn Ontology

We have analyzed many existing recipe datasets to extract desired information and knowledge for the development of *RecipeOn* ontology, none of the existing resources could cover all the requirements of the proposed ontology (for example Recipe1M+ (Marin et al., 2019) lacks cuisine, time, and nutritional information; RecipeNLG (Bień et al., 2020) lacks the data about nutrition; RecipeDB (Batra et al., 2020) and food.com dataset (Majumder et al., 2019) covers relatively more attributes but it lacks the heterogeneity and is relatively smaller). We have analyzed around one million recipes data manually & computationally and devised the following fulfilling criteria for *RecipeOn*.

TABLE 4.1: RecipeOn' Ontology Requirements Specification Document (ORSD).

1. Purpose
RecipeOn has been developed to semantically understand the recipe as a process and describe its modules using primary and secondary attributes. RecipeOn should also describe and support the recipe generation process using alternative ingredients and actions.

2. Scope
RecipeOn will focus on the food recipe creation and novel recipe generation process.

3. Implementation
RecipeOn has been implemented using OWL in Protege (version 5.5).

4. Intended Users
A. Sick Persons
B. Diet Conscious People
C. Food Enthusiast

5. Intended Uses
A. To find suitable recipes that are made using the ingredients at hand.
B. To find the recipes that avoid certain ingredients or actions (such as vegetarian recipes for gout patients or non-fried recipes for heart patients).
C. To find recipes that have high or low concentrations of some specific nutrient.
D. To generate novel recipes by replacing certain actions or ingredients with alternative actions or ingredients.
E. To generate a novel recipe by replacing a sub-procedure (set of actions) with an alternative sub-procedure.

6. Non Functional Requirements

Table 4.1: RecipeOn' Ontology Requirements Specification Document (ORSD). Continued

The only non-functional requirement of RecipeOn is re-usability. We want to reuse the maximum possible ontological and non-ontological resources. Re-usability of ontology design patterns is also the primary focus of RecipeOn.

7. Functional Requirements

Q_c1 ***Core-Recipe Related Competency Questions***

$Q_c1.1$ I am feeling sick and low energy. Let me know which **soup** recipes can be made in **30 minutes** (i.e quickly or in less time)?

$Q_c1.2$ What are **diabetic friendly** recipes?

$Q_c1.3$ Which are high **protein vegetarian recipes**?

$Q_c1.4$ Which are the appropriate recipes for **eggeterians**?

$Q_c1.5$ Which **Italian** recipes can be made **easily**?

$Q_c1.6$ Which **breakfast** recipes are **rated** more than 1000 times(i.e mostly rated)?

$Q_c1.7$ Which **appetizers** can be made using **oven**?

Q_c2 ***Ingredients Related Competency Questions***

$Q_c2.1$ Which recipes contain **vegetables** and exclude **red meat & seafood** (suitable for gout patients)?

$Q_c2.2$ What are **non milk** (gluten free) recipes?

$Q_c2.3$ What are **wheat free** (non celiac) recipes?

$Q_c2.4$ Which deserts uses sugar in low amount (for pre-diabetic persons)?

$Q_c2.5$ Which recipes contain leafy greens (suitable for reducing belly fat)?

$Q_c2.6$ Which rice recipes have less number of spices?

$Q_c2.7$ Which recipes use fish as the main ingredient?

$Q_c2.8$ I have chicken, yogurt, spices, & olive oil at hand. Which recipes can be made using these available ingredients?

$Q_c2.9$ Which ingredients are needed for pulao?

Q_c3 ***Actions Related Competency Questions***

$Q_c3.1$ What are the **non fried** recipes?

$Q_c3.2$ Which are **boiled** recipes?

$Q_c3.3$ Which **fried** recipes can I make using the fryer?

Q_c4 ***Nutritions Related Competency Questions***

$Q_c4.1$ Which recipes have **low cholesterol** (suitable for persons that have bad lipid profile)?

$Q_c4.2$ What are most suitable **iron rich** recipes (for anemic persons) ?

$Q_c4.3$ Which recipes are **rich in vitamin c**[1] (for immunity boosting)?

$Q_c4.4$ What are **protein rich** recipes?

$Q_c4.5$ What are **fiber rich**[11] recipes?

$Q_c4.6$ What are **low calories**[11] recipes?

Q_c5 ***Procedures Related Competency Questions***

$Q_c5.1$ Which recipes contain **baked chicken**?

$Q_c5.2$ Which recipes contain **boiled noodeles**?

$Q_c5.3$ What recipes contain **fried chicken**?

Table 4.1: RecipeOn' Ontology Requirements Specification Document (ORSD). Continued

8.	**Glossary of Terms** Recipe, Ingredient, Action, Procedure, Nutrition, Cooking Time, Protein, Fat, Calories, Cholesterol, Oven, Frying, Cooking Time, boil, appetizers, Vegetable, Cuisine, Course, Aggregate Rating, Preparation Actions, Cutting, Shred, Potato, Chicken, Sugar, Lemon, Vinegar, Lime, Cold Coffee, Baking, Fryer, Yogurt, Olive Oil, Leafy Green, Spice

4.1.1 Requirements of RecipeOn

The important requirements for RecipeOn ontology can be divided into functional and non-functional requirements discussed in the following subsections.

4.1.1.1 Functional Requirements of RecipeOn

RecipeOn has a diverse set of competency questions that formulate its functional requirements. After careful analysis of these competency questions, we grouped them into five different Components. Each component maps a unique dimension of RecipeOn ontology. These components include Core-Recipe, Ingredients, Actions, Nutrition, and Procedures. To the best of our knowledge, none of the existing ontologies address the key competency questions related to these five modules, as mentioned in Table 4.1. We have analyzed several ontological and non-ontological resources to fine-tune the functional requirements of $RecipeOn$ ontology.

- *Core-Recipe*: Core-recipe defines the basic characteristics of a recipe. These include cooking time, diet, cuisine, meal, or cooking style of a recipe, There are a few ontological resources that list a small number of properties that provides basic information about a recipe. FoodOn[2] (Dooley et al., 2018), BBC Food Ontology (BBC, 2014), An analytical approach to building a

[2]https://www.ebi.ac.uk/ols/ontologies/foodon

core ontology for food (Madalli et al., 2017), Food Track & Trace Ontology (FTTO) (Pizzuti et al., 2014), An Ontology Design Pattern for Cooking Recipes - Classroom Created (Sam et al., 2014), and Cooking Ontology (Batista et al., 2006) are few of these resources. Many basic properties (such as rating count, review count, course name, meal type, cooking method, date published, recipe yield, yield, and tools) are still missing in these ontologies. Despite being comprehensive, existing ontologies do not cover the required scope of the core-recipe module as mentioned under competency questions Q_c1 in Table 4.1. For example, these ontologies are unable to find soup recipes that can be prepared in 30 minutes or can't find Italian recipes that can be made easily.
We have also explored non-ontological resources like classification schemes, databases, glossaries, and thesauri (Villazón-Terrazas et al., 2010). Major search engines (like Google, Bing, and Yahoo) encourage recipe-related websites to follow Recipe Schema[3]. The schema helps search engines to understand the meanings of recipe-related text. $RecipeSchema$ lists many properties that can be included in recipe-related ontologies, especially in the core-recipe module, to fulfill the scope requirements. RecipeOn connects and reuses these core properties from $RecipeSchema$ as much as required.

- *Ingredients*: There are several ontologies available that cover ingredients, including FIDEO[4] (Food Interactions with Drug Evidence Ontology) (Bordea et al., 2020), FOBI[5] (Food-Biomarker) (Castellano-Escuder et al., 2020), Food ontology model for a healthcare service (Lee, 2012), Food product ontology (Kolchin and Zamula, 2013), Food system resilience (Tendall et al., 2015). These ontologies define ingredient class hierarchy based on the botanical nature of the ingredients. This classification is nonsuitable to understand the role of the ingredient in the cooking process.
Recipe1m+ (Marin et al., 2019) shares a list of ingredients. BBC Food[6] presents a large number of ingredients along with their brief description

[3] https://schema.org/Recipe
[4] https://bioportal.bioontology.org/ontologies/FIDEO
[5] https://bioportal.bioontology.org/ontologies/FOBI
[6] https://www.bbc.co.uk/food/ingredients

and their usage. These resources also don't define an appropriate ingredient classification considering the ingredient's role in a recipe. RecipeOn's ingredient class hierarchy is primarily inspired by the ingredient taxonomy mentioned at The Cooks Thesaurus[7].

- *Actions*: There are a few ontologies related to cooking actions (i.e, ontology (Ribeiro et al., 2006), ontology construction:cooking domain (Batista et al., 2006), and action ontology (Antô et al., 2020)). Among these ontologies, action ontology is discussed in more detail. It classifies actions into three main categories and several subcategories. The authors have also discussed the evolution of recipes.

  The non ontological resources include the food service taxonomy (Pereira et al., 2022) that also includes the cooking action taxonomy. The categorization of actions into classes and subclasses is interesting and is a close fit to the cooking process in reality.

  The ontological and non-ontological resources discussed above are not suitable to represent a recipe as a process as they do not represent the sequence in which actions are being executed. Moreover, none of the above-mentioned ontologies categorize the actions in a way that may help in the new recipe generation process to generate novel recipes (discussed in more detail under *Alternative Ingredient/Action* and *Alternative Procedure*).

- *Nutrition*: Many ontological resources are available for nutrition. In a comparison of ontological resources (including FoodWiki (Çelik, 2015), Helis (Dragoni et al., 2018), PerkApp (Bailoni et al., 2016), OFFF (Ontology of Fast Food Facts) (Amith et al., 2020), ontology for managing diets of hypertensive individuals (Clunis, 2019), nutrition schema[8], food & nutrition ontology for Thai pre school (Lertkrai et al., 2018), and the ontology of fast food facts (Amith et al., 2021), we have selected schema.org for the nutrition module of RecipeOn as it is simple and uses nutrient names as properties to map recipes to the corresponding quantities (Scenario 4 of

[7] http://www.foodsubs.com/
[8] https://schema.org/NutritionInformation

NeOn).

- *Procedure*: While we could not find ontologies covering the cooking procedures, Execution-Executor-Procedure[9] (EEP) ontological design pattern relates to the general/construction domain defines procedure concepts. These concepts can be combined (Scenario 4 of NeOn) with the non-ontological resource discussed in AutoChef (Jabeen et al., 2020) to fulfill the procedure requirements for recipes (Scenario 2 of NeOn).

4.1.1.2 Non-Functional Requirements of RecipeOn

We have designed RecipeOn to maximize the reusability of the classes and properties. The aim of this reusability is twofold. Firstly we want to maximize the reuse of existing classes and properties while designing RecipeOn. Secondly, we want to make sure that RecipeOn and its components can be easily reused by other ontologies. To achieve the goals, we have performed the following tasks.

- We have used classes and properties from existing ontological design patterns as much as possible. And only designed/extended our classes where it was necessary.

- We designed flexible concepts and properties with less ontological commitments/restrictions. This flexibility makes it easy for other ontologies to reuse concepts/properties from RecipeOn.

- To increase the findability and thus reusability of RecipeOn, we have placed RecipeOn ontology along with its documentation[10] at Github[11].

[9] http://ontologydesignpatterns.org/wiki/Community:Building_and_Construction
[10] created using Live OWL Documentation Environment (LODE) https://essepuntato.it/lode/
[11] https://raw.githubusercontent.com/HajiraJabeen/EvoRecipesOntology/main/EvoRecipes.owl

TABLE 4.2: Re usability of Classes, Object properties, and Data properties from various Namespaces to RecipeOn Ontology.

Prefix	Ontology Namespace	Classes	Object Prop.	Data Prop.
owl	Web Ontology Language[12]	04	01	-
schema	Schema[13]	04	17	14
evo	RecipeOn[14]	193	43	03
seq	Sequence[15]	-	04	-
qudt	Quantity, Unit, Dimension, and Type[16]	03	02	02
	Total	**204**	**67**	**19**

4.2 Ontology Overview

The RecipeOn ontology is divided into five modules and each module is capable to answer a specific set of competency questions. These modules are capable to answer simple competency questions like finding ingredient-specific recipes(e.g. fish recipes) to complex competency questions like finding recipes that can be made using ingredients at hand. A description of each module is presented in sections 4.2.1 - 4.2.5.

RecipeOn consists of 231 classes, 37 object properties, and 20 data properties. The reusability (of classes, object properties, & data properties) is the primary focus of *RecipeOn*. A summary of the components from different namespaces reused in *RecipeOn* is presented in Table 4.2.
To test the modules and the consistency with the functional requirements,

We have formulated a few SPARQL queries presented in Tables 4.3, 4.4, 4.5, and 4.6 against each competency question mentioned in Table 4.1.

[11] https://www.webmd.com/food-recipes/guide/vitamins-and-minerals-good-food-sources
[12] http://www.w3.org/2002/07/owl
[13] http://www.schema.org
[14] https://hajirajabeen.github.io/EvoRecipesOntology/
[15] http://ontologydesignpatterns.org/cp/owl/sequence.owl
[16] http://qudt.org/2.1/schema/qudt

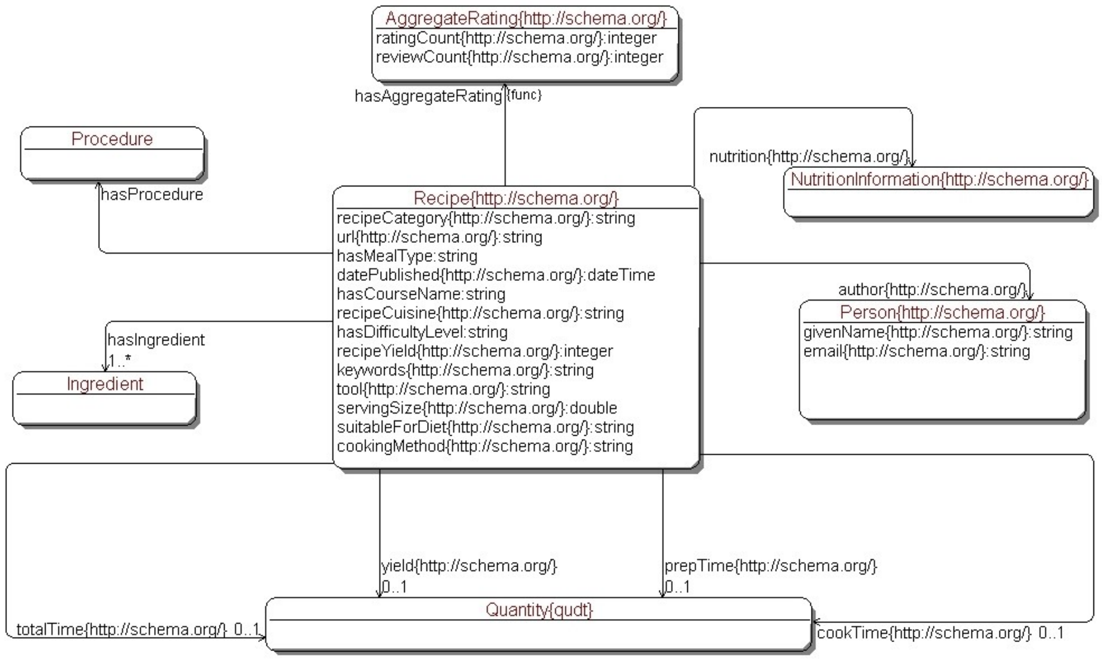

FIGURE 4.1: Classes and Properties Involved in Core-Recipe Module.

4.2.1 Core-Recipe

Core-recipe is the basic module of RecipeOn ontology and intends to describe the fundamental characteristics of a recipe. There are several key properties desired for a recipe. These properties (`schema:recipeCuisine`, `schema:recipeCourse`, `schema:recipeYield`, `mealType`, `schema:yield`, `schema:keywords`, `schema:author`, `schema:recipeYield`, `schema:suitableForDiet` etc) could help to depict the recipe in more detail. Not only these factors are interesting for intended users but these factors could also be helpful to better categorize the recipes.

`schema:Recipe` is the main class of this module and has the maximum number of object and data properties compared to any other class of RecipeOn ontology. Figure 4.1 (created using OWLGrEd[17]) shows the main classes and properties involved in core-recipe. `IngredientType` and `Procedure` are connected with `schema:Recipe` through specialized version of `hasPart` property of `PartOf`[18] namespace.

[17] http://owlgred.lumii.lv/online_visualization
[18] http://ontologydesignpatterns.org/wiki/Submissions:PartOf

Core-recipe module is capable to answer the competency questions mentioned under Q_c1. Table 4.3 shows the SPARQL query against each of these competency questions.

TABLE 4.3: SPARQL Queries against Competency Questions Q_c1 mentioned in Table 4.1.

$Q_c1.1$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe <http://schema.org/recipeCategory> ?cat .
?recipe <http://schema.org/totalTime> ?time .
?time qudt:hasQuantityValue ?qValue .
?qValue qudt:hasNumericValue ?nv .
?time qudt:hasUnit ?u .
?u qudt:Abbreviation ?a
FILTER (?nv <= "30"^^xsd:double)
FILTER(?cat="soup"^^xsd:string)
}
```

$Q_c1.2$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe <http://schema.org/suitableForDiet> ?diet .
FILTER(?diet="diabetic friendly"^^xsd:string)
}
```

SPARQL queries against Q_c1 competency questions mentioned in Table 4.1 {Continued}.

$Q_c1.3$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe <http://schema.org/suitableForDiet> ?diet .
FILTER(?diet="protein vegetarian"^^xsd:string)
}
```

$Q_c1.4$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe <http://schema.org/suitableForDiet> ?diet .
FILTER(?diet="eggeterian"^^xsd:string)
}
```

$Q_c1.5$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe <http://schema.org/recipeCuisine> ?cuisine .
?recipe :hasDifficultyLevel ?difficulty
FILTER(?cuisine="italian"^^xsd:string)
FILTER(?difficulty="easy"^^xsd:string)
}
```

SPARQL queries against Q_c1 competency questions mentioned in Table 4.1 {Continued}.

$Q_c1.6$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe <http://schema.org/ratingCount> ?rating .
?recipe :hasMealType ?meal
FILTER(?rating>="1000"^^xsd:string)
FILTER(?meal="breakfast"^^xsd:string)
}
```

$Q_c1.7$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe <http://schema.org/tool> ?usedTool .
?recipe :hasCourseName ?course
FILTER(?usedTool>="oven"^^xsd:string)
FILTER(?course="appetizer"^^xsd:string)
}
```

4.2.2 Ingredients

The ingredient can be defined as "any of the foods or substances that are combined to make a particular dish". Ingredients play a vital role in a recipe for their nutritional, functional, and sensory characteristics. They also add flavor, color, or texture to the food. We have classified the ingredients into 12 classes and multiple sub-classes as shown in Figure 4.2. This classification is based

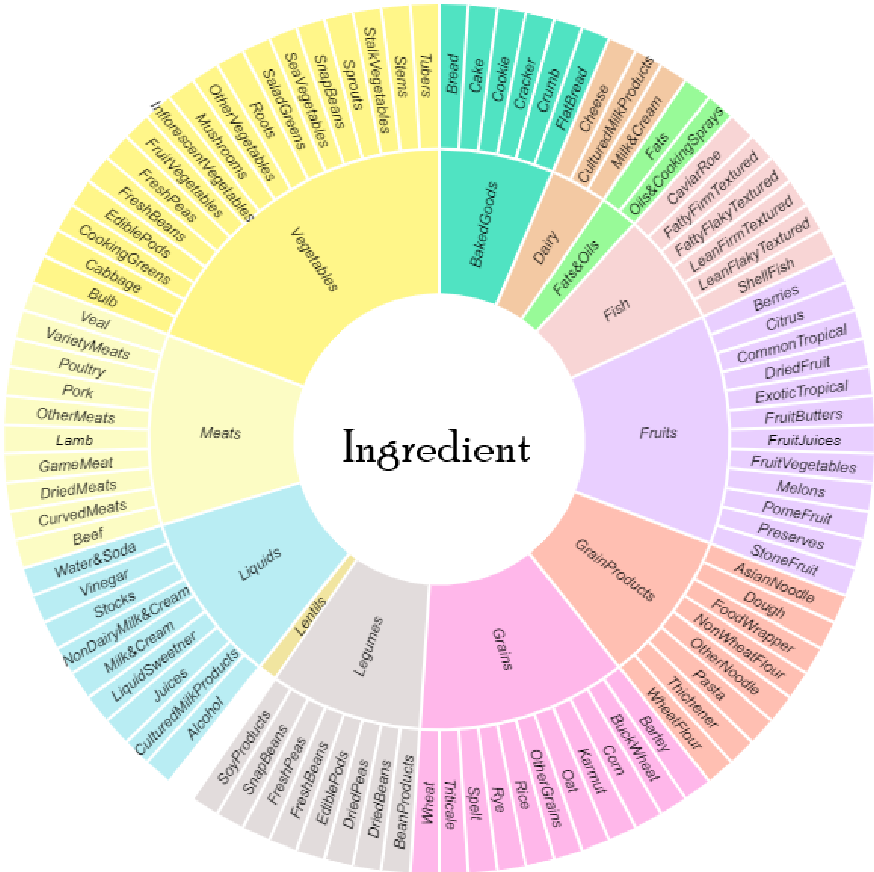

FIGURE 4.2: Ingredients Class Hierarchy

on the cook thesaurus[7] categorization. Moreover, each ingredient has a specific role in the cooking process that may vary across different recipes as main, side, or flavor. So we have also labeled ingredients using `IngredientType` class to identify the role of an ingredient in a recipe. For example tomato is a main ingredient in "Tomato Ketchup[19]" but a side ingredient in "Chicken Korma[20]"

In our recipe concept model, the ingredients module has the following relationships as shown in Figure 4.3.

- An `Ingredient` can have a substitute of type `Ingredient`.
- An `Ingredient` always has `qudt:Quantity`.
- An `Ingredient` has an `owl:Action` performed on it.
- An `Ingredient` belongs to a type `IngredientType`.

[19] https://www.vegrecipesofindia.com/tomato-ketchup-recipe-tomato-sauce/
[20] https://www.teaforturmeric.com/authentic-chicken-korma/

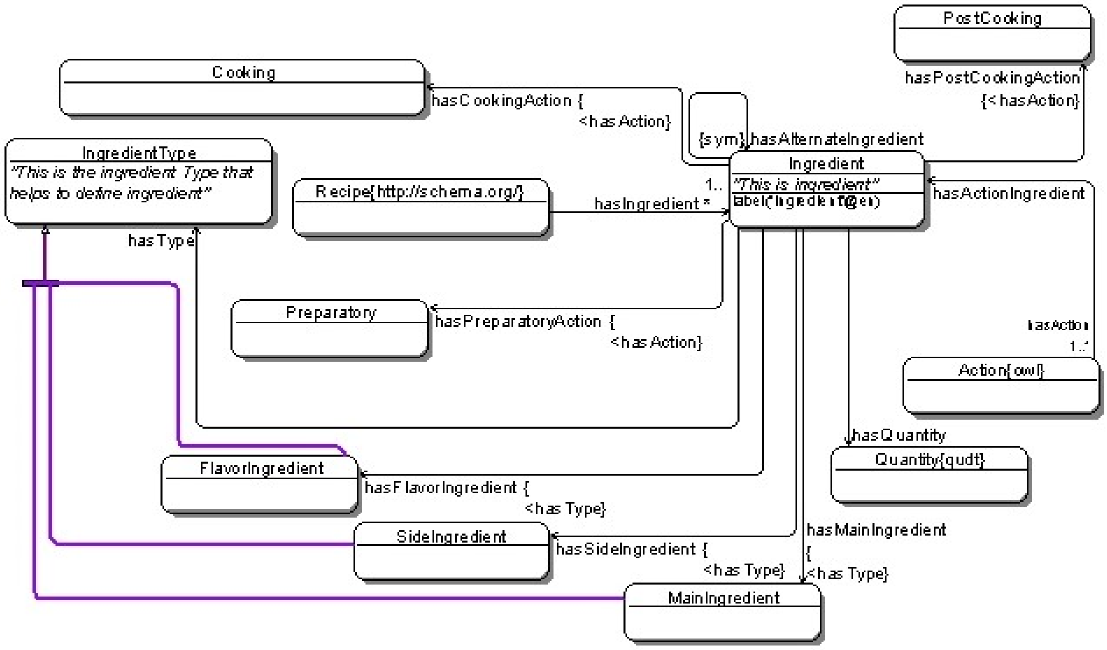

FIGURE 4.3: Object Properties and Classes related to Ingredient Module.

- `IngredientType` has three sub classes i.e `MainIngredient`, `SideIngredient`, and `FlavoringIngredient`.
- `Ingredient` links with `MainIngredient`, `SideIngredient`, and `FlavoringIngredient` through sub properties `hasMainIngredient`, `hasSideIngredient`, and `hasFlavoringIngredient` respectively.
- An `Ingredient` has an alternate/substitute of type `Ingredient` defined using the relationship `hasAlternateIngredient`.
- Every `Ingredient` has an object property `hasAction` that defines the `Action` performed on an `Ingredient`.
- `Ingredient` links with subclasses of `Action` i.e `Preparatory`, `Cooking`, and `PostCooking` through `hasPreparatoryAction`, `hasCookingAction`, and `hasPostCookingAction` respectively.

The ingredients module can handle all competency questions mentioned under Q_c2 in Table 4.1. SPARQL queries against each of these questions are shown in Table 4.4.

TABLE 4.4: SPARQL Queries against Competency Questions Q_c2 mentioned in Table 4.1.

$Q_c2.1$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT DISTINCT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient ?ingredient
FILTER EXISTS {
?ingredient a :Vegetable . }
FILTER NOT EXISTS{
?ingredient a :Meat .
?ingredient a :Lamb .
?ingredient a :Fish . }
}
```

$Q_c2.2$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT DISTINCT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient ?ingredient .
FILTER NOT EXISTS{
?ingredient a :Dairy  }
}
```

SPARQL queries against Q_c2 competency questions mentioned in Table 4.1 {Continued}.

$Q_c2.3$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT DISTINCT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient ?ingredient .
FILTER NOT EXISTS{
?ingredient a :Wheat  }
}
```

$Q_c2.4$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT DISTINCT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient :BrownSugar .
}
```

$Q_c2.5$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT DISTINCT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient ?ingredient .
?ingredient a :CookingGreens .
}
```

SPARQL queries against Q_c2 competency questions mentioned in Table 4.1 {Continued}.

$Q_c2.6$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT DISTINCT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient :Rice .
?recipe :hasIngredient ?ingredient .
FILTER NOT EXISTS{
?ingredient :hasFlavoringIngredient ?flavor .   }
}
```

$Q_c2.7$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT DISTINCT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient :Fish .
:Fish :hasMainIngredient :MainIngredient .
}
```

SPARQL queries against Q_c2 competency questions mentioned in Table 4.1 {Continued}.

$Q_c2.8$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT DISTINCT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient :Chicken .
?recipe :hasIngredient :Yogurt .
?recipe :hasIngredient :BlackPepper .
?recipe :hasIngredient :OliveOil .
}
```

$Q_c2.9$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT DISTINCT ?ingredient
WHERE { :Pulao a <http://schema.org/Recipe> .
:Pulao :hasIngredient ?ingredient .
}
```

4.2.3 Actions

Action module represent the set of small purposeful activities that are performed on various ingredients while cooking. The module includes simple activities like washing an ingredient to complex activities like mixing a set of ingredients. The higher the granularity of these actions, the more detailed information is required to define their appropriate relationships with ingredients (e.g., you can chop potatoes but you cannot chop rice) and other actions (e.g.

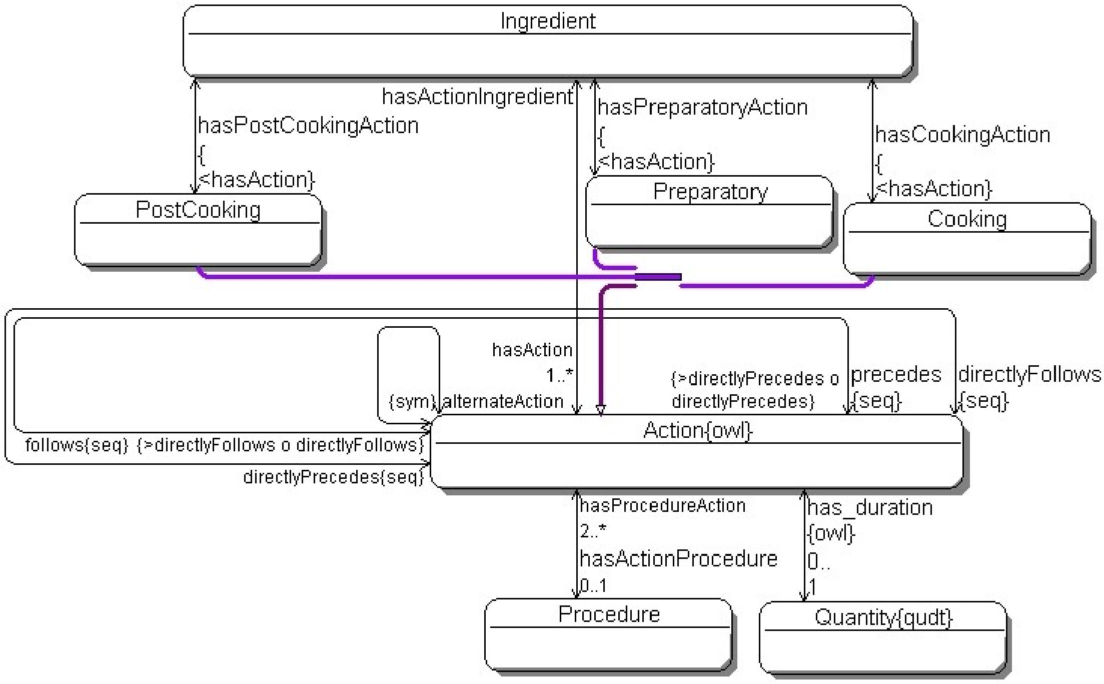

FIGURE 4.4: Object Properties and Classes Related to Action Module.

you can boil water and then may combine it with other ingredients but you can't sprinkle water after boiling). Actions may also have some additional constraints attached to them (such as stirring mixture until thick; heating milk up to 46°F; Cutting potatoes into 1-inch length strips; Fry chicken for 5 minutes).

The actions module also maintains a sequence of activities as opposed to other modules of RecipeOn. A sequence is mandatory due to the reason that activities without a sequence may not produce the desired outcome. This sequence is normally represented on online recipe websites using numbered or bullet-based instructions. To handle the sequence in actions, we have used `seq:directlyFollows` and `seq:directlyPrecedes` as follow.

```
Class: owl:Action

ObjectProperty: seq:directlyPrecedes
    Domain:
       owl:Action
    Range:
       owl:Action
    InverseOf:
       seq:directlyFollows
```

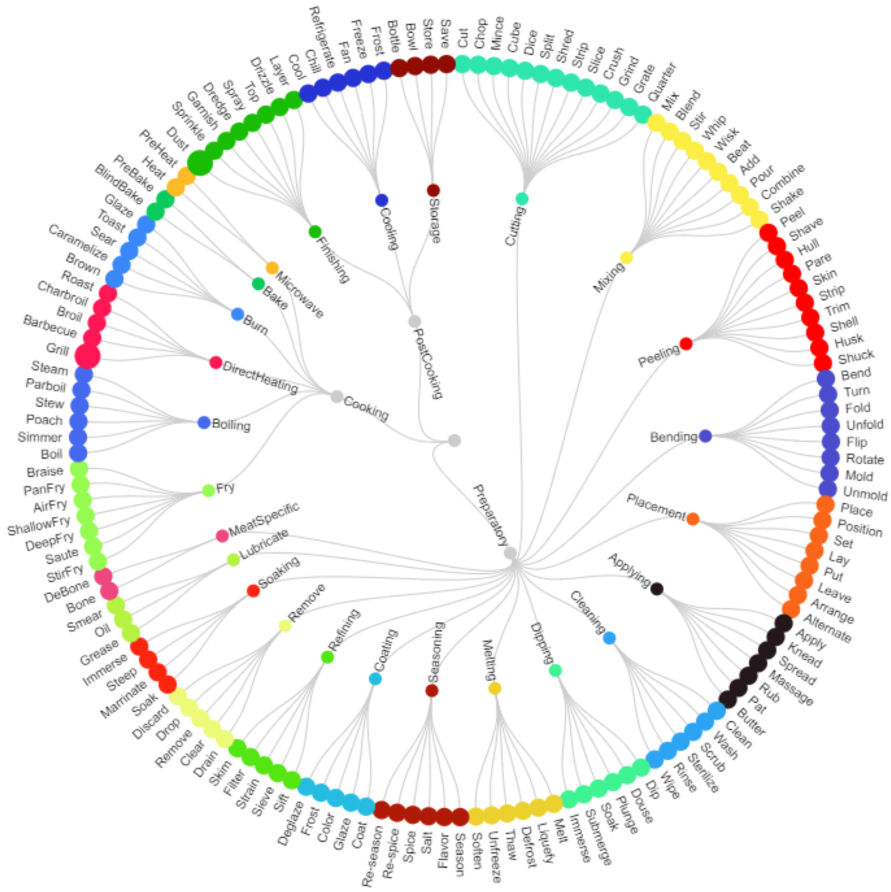

FIGURE 4.5: Hierarchy of Classes, Sub-Classes, and Individuals in the Action Module.

`owl:Action` is the main class of *Actions* module and represents an individual activity. `owl:Action` has three sub classes (i.e. `Cooking`, `Preparatory`, and `PostCooking`) attached to it using `rdfs:subClassOf` object property as shown in Figure 4.4. A more detailed hierarchy of `owl:Action` subclasses and individuals that lie in each subclass has been shown in Figure 4.5. `owl:Action` maintains the sequence of actions using `seq:directlyFollows` and `seq:directlyPrecedes` object properties as shown in the above listing. Both properties are inverse to each other. `owl:Action` expresses the quantifiable assertions using object properties (like `owl:has_duration`, `hasTemperature`) with `qudt:Quantity`. On the other hand for non quantifiable assertions (e,g. rinse with water until clean), `owl:Action` uses data property `until`.

The actions module can handle all competency questions mentioned under Q_c3 in Table 4.1. SPARQL queries against each of these questions are shown in Table 4.5.

TABLE 4.5: SPARQL Queries against Competency Questions Q_c3 mentioned in Table 4.1.

$Q_c3.1$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient ?ingredient .
?ingredient a :Ingredient .
FILTER NOT EXISTS {
?ingredient :hasAction :fry }
}
```

$Q_c3.2$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient ?ingredient .
?ingredient :hasAction :Boil
}
```

SPARQL queries against Q_c3 competency questions mentioned in Table 4.1 {Continued}.

$Q_c3.3$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe <http://schema.org/tool>?tool .
?recipe :hasIngredient ?ingredient .
?ingredient :hasAction :Fry
FILTER(?tool="Fryer"^^xsd:string)
}
```

4.2.4 Nutrition

Nutrition presents nutrient information related to a recipe. The nutrient information of a recipe is the sum of nutrients present in individual ingredients. Nutrition information is potentially interesting and useful for sick and diet-conscious people.

`schema:NutritionInformation` is the main class of *Nutrition* module. There are 30+ food nutrients that are represented using nutrient class as shown in Figure 4.6. Amount of the nutrient is expressed using `qudt:Quantity` This module has the ability to answer the competency questions mentioned under Q_c4 in Table 4.1. Table 4.6 lists all the SPARQL queries against each of these competency questions.

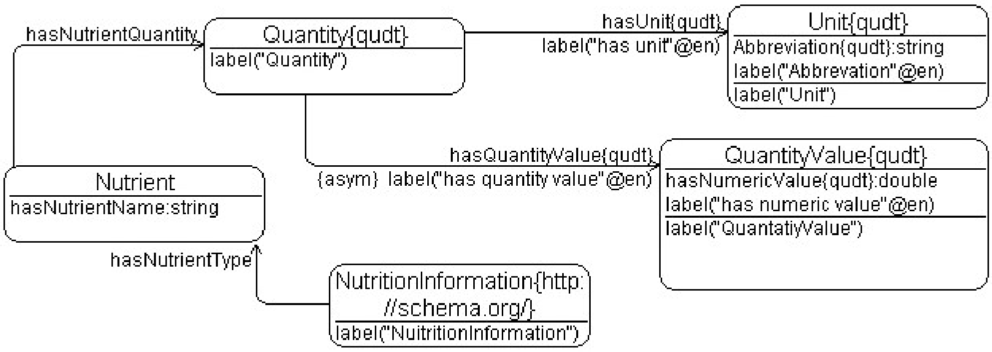

FIGURE 4.6: Object Properties and Classes related to Nutrition Module.

TABLE 4.6: SPARQL Queries against Competency Questions Q_c4 mentioned in Table 4.1.

$Q_c4.1$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasNutrientType :CholestrolContent.
:CholestrolContent a :Nutrient.
:CholestrolContent :hasNutrientQuantity ?quan .
?quan qudt:hasQuantityValue ?qv .
?qv qudt:hasNumericValue ?nv
FILTER (?nv <= "2"^^xsd:double)
}
```

SPARQL queries against Q_c4 competency questions mentioned in Table 4.1 {Continued}.

$Q_c4.2$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasNutrientType :Iron.
:Iron a :Nutrient.
:Iron :hasNutrientQuantity ?quan .
?quan qudt:hasQuantityValue ?qv .
?qv qudt:hasNumericValue ?nv
FILTER (?nv >= "1.5"^^xsd:double)
}
```

$Q_c4.3$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasNutrientType :VitaminC.
:VitaminC a :Nutrient.
:VitaminC :hasNutrientQuantity ?quan .
?quan qudt:hasQuantityValue ?qv .
?qv qudt:hasNumericValue ?nv
FILTER (?nv >= "26"^^xsd:double)
}
```

SPARQL queries against Q_c4 competency questions mentioned in Table 4.1 {Continued}.

$Q_c4.4$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasNutrientType :Protein.
:Protein a :Nutrient.
:Protien :hasNutrientQuantity ?quan .
?quan qudt:hasQuantityValue ?qv .
?qv qudt:hasNumericValue ?nv
FILTER (?nv >= "5"^^xsd:double)
}
```

$Q_c4.5$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasNutrientType :FiberContent.
:FiberContent a :Nutrient.
:FiberContent :hasNutrientQuantity ?quan .
?quan qudt:hasQuantityValue ?qv .
?qv qudt:hasNumericValue ?nv
FILTER (?nv >= "2"^^xsd:double)
}
```

SPARQL queries against Q_c4 competency questions mentioned in Table 4.1 {Continued}.

$Q_c4.6$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasNutrientType :CalorieContent.
:CalorieContent a :Nutrient.
:CalorieContent :hasNutrientQuantity ?quan .
?quan qudt:hasQuantityValue ?qv .
?qv qudt:hasNumericValue ?nv
FILTER (?nv <= "200"^^xsd:double)
}
```

4.2.5 Procedures

The procedure is like a sub-recipe and represents a set of relevant activities that may be carried out on a subset of ingredients to prepare an intermediate ingredient or to complete a subtask of a recipe. For example "Homemade Apple Pie"[21] represents three procedures (i.e Make the pie crust; Make the apple pie filling; Assemble the apple pie). Similarly "Paneer Patties"[22] represents three procedures (i.e Boiling potatoes; Making paneer patties; Frying paneer patties). Object properties and classes related to the procedure module are shown in Figure 4.7.

`Procedure` is the main class of procedures module. Similar to `Action`, `Procedure` also has a sequence of `Procedure` as shown in the following listing.

```
Class: Procedure

```

[21] https://www.acozykitchen.com/a-classic-apple-pie

[22] https://www.vegrecipesofindia.com/aloo-paneer-tikki/

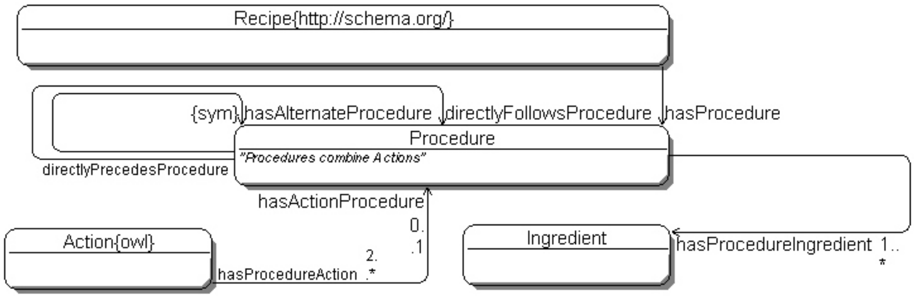

FIGURE 4.7: Object Properties and Classes Related to Procedures Module.

```
ObjectProperty: seq:directlyPrecedesProcedure
    Domain:
       Procedure
    Range:
       Procedure
    InverseOf:
       seq:directlyFollowsProcedure
```

`Procedure` uses `hasProcedureAction` and `hasProcedureIngredient` properties to link with `Action` and `Ingredient` respectively. A `Procedure` can also replace itself using `hasAlternateProcedure` object property. Moreover, the procedure module answers the competency questions mentioned under Q_c5 in Table 4.1. Table 4.7 lists all the SPARQL queries against each of these competency questions.

TABLE 4.7: SPARQL Queries against Competency Questions Q_c5 mentioned in Table 4.1.

$Q_c5.1$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient :Chicken .
:Chicken :hasAction :Bake .
}
```

SPARQL queries against Q_c5 competency questions mentioned in Table 4.1 {Continued}.

$Q_c5.2$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient :Noodles .
:Chicken :hasAction :Boil .
}
```

$Q_c5.3$

```
PREFIX : <https://hajirajabeen.github.io/
    EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient :Chicken .
:Chicken :hasAction :Fry .
}
```

4.2.6 Property Restrictions

Classes are connected to other classes and literal values through object and data properties, but these relationships may have restrictions on them. These restrictions should be satisfied for the existence of a property and can be classified into universal, existential, or cardinality restrictions. Universal restriction (`owl:allValuesFrom`) specifies that the restriction should be satisfied for every member of the class. On the other hand existential restriction (`owl:someValuesFrom`) specifies that it should be satisfied for some members of the class. Cardinality restrictions are further divided into min (`owl:minQualifiedCardinality`), max (`owl:maxQualifiedCardinality`), and exact (`owl:qualifiedCardinality`) cardinality types.

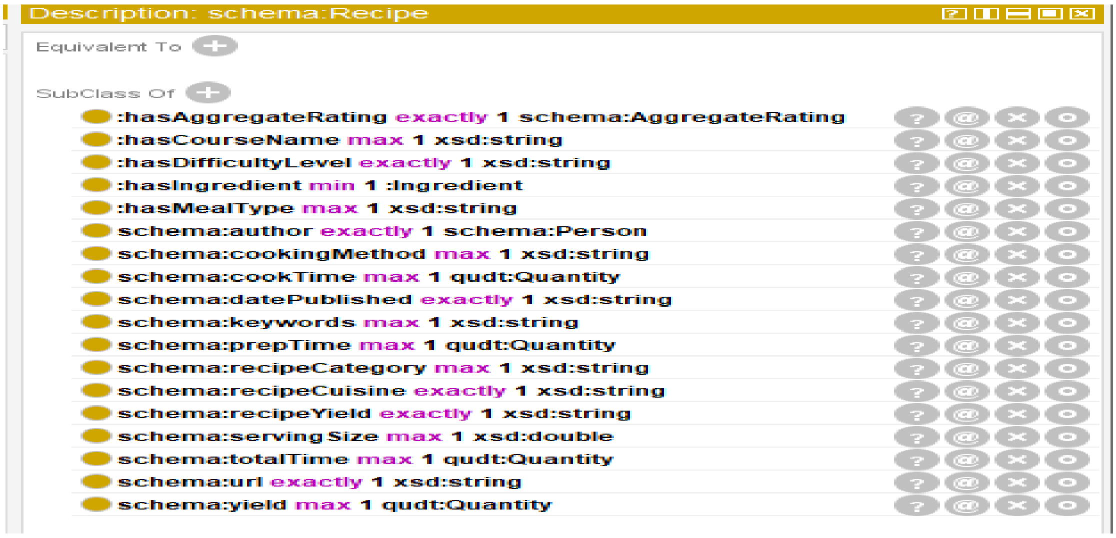

FIGURE 4.8: Recipe

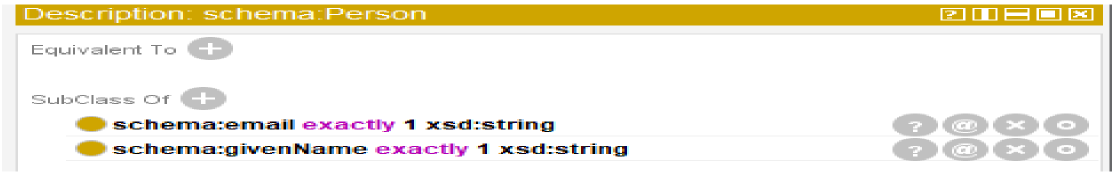

FIGURE 4.9: Person

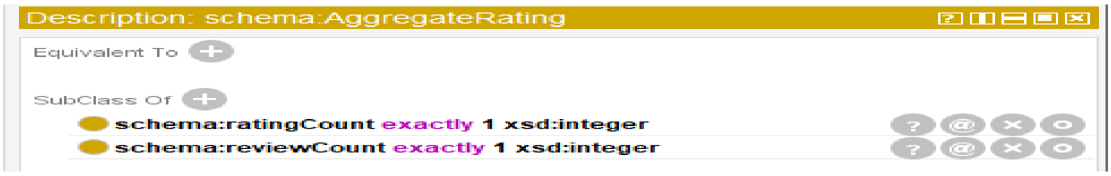

FIGURE 4.10: AggregateRating

Figures 4.8, 4.9, 4.10 shows the property restrictions for main classes (i.e, Recipe, Person, and AggregateRating) of the core-recipe module. The recipe should have at least one ingredient. The recipe has few mandatory properties (i.e hasAggregateRating, hasDifficultyLevel, author, datePublished, recipeYield, and url) and a few optional properties (i.e hasCourseName, hasDifficultyLevel, hasMealType, cookingMethod, cookTime, keywords, prepTime, recipeCategory, recipeCuisine, totalTime, and recipeYield). Person class has a mandatory attribute (givenName) and an optional attribute (email), while AggregateRating has two mandatory attributes ratingCount and reviewCount.

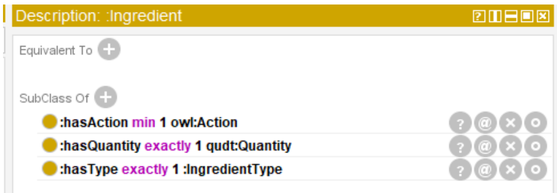

Figure 4.11: Excerpt of Ingredient Class

In the Ingredients module we have defined property restrictions on ingredient class as shown in Figure 4.11. Every ingredient should have at least one action that should be performed on it. Moreover, each ingredient should have exactly one quantity and an ingredient type.
Property restrictions on the Action class specify that each action should either be performed on an ingredient or on top of another action. An action can be part of a procedure and each action may have a time duration associated with it. These restrictions are listed in Figure 4.12.

Figure 4.12: Excerpt of Action Class.

Nutrition module in Figures 4.13, 4.14, 4.15, 4.16 lists a qualified cardinality restriction defined using hasNutrientQuantity on Nutrient Class. In addition, there are few qualified cardinality restrictions defined using hasQuantityValue, hasUnit, and hasNumericVlaue while Abbreviation is defined using max qualified cardinality.

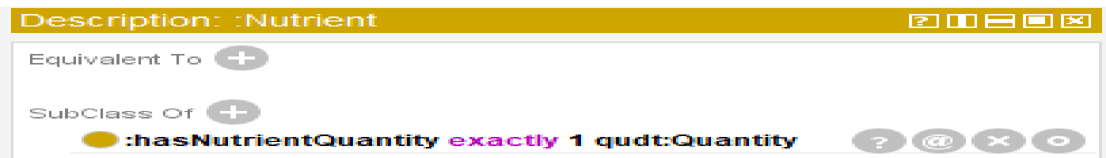

Figure 4.13: Nutrient

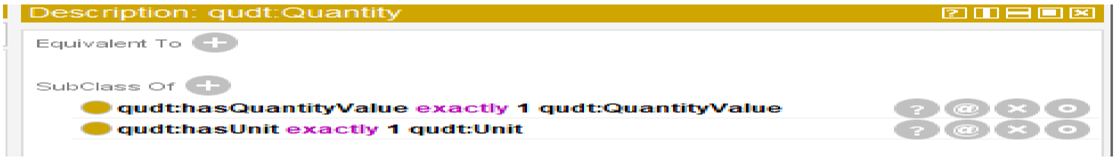

FIGURE 4.14: Quantity

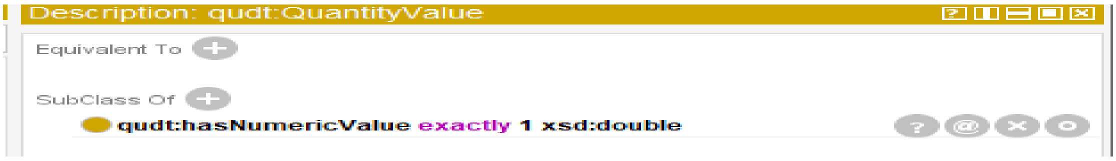

FIGURE 4.15: QuantityValue

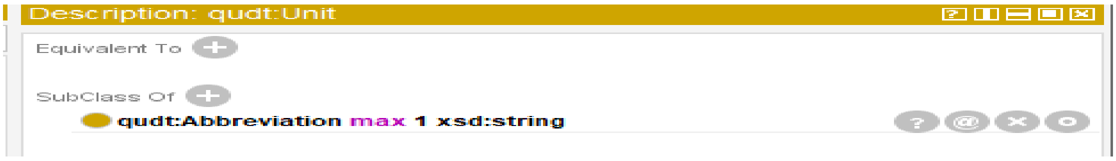

FIGURE 4.16: Unit

We have defined two property restrictions on procedure class as shown in Figure 4.17. These specify that each procedure should have at least 2 actions and at least one ingredient.

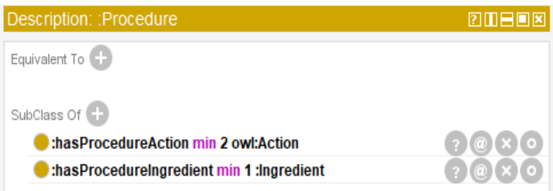

FIGURE 4.17: Excerpt of Procedure Class.

4.3 Ontology Evaluation

Real-world ontologies are complex to handle as they have thousands of concepts involved in them. Ontology evaluation is an important factor in making ontology consistent. RecipeOn is evaluated through Ontology Pitfall Scanner (OOPS) (Poveda-Villalón et al., 2014) and (Garijo, 2017), Ontology quality parameters as prescribed in (Vrandečić, 2009). Furthermore, the Ontology metrics are summarized in section 4.3.3.

4.3.1 OOPS Evaluation

OOPS(OntOlogy Pitfall Scanner) pitfall scanner is a set of rules that helps you find the common pitfalls that may occur while developing ontologies. RecipeOn was scanned using a web-based OOPS scanner[23]. Few reported important and critical pitfalls were fixed to improve the quality of RecipeOn ontology.

4.3.2 Ontology Quality

4.3.2.1 Accuracy

We have used top recipe websites[24] to understand the recipe process, its components (i.e ingredients, actions, nutrition, and procedures), and it's representation to develop the RecipeOn ontology. RecipeOn ontology uses some of its components(i.e. classes, object properties, and data properties) from recipe schema[25]. Also it reuses some of the components from existing ontology design patterns including Seq[26](to represent sequence of actions i.e. directlyFollows & directlyPrecedes) and QUDT[27](to represent quantities i.e. Quantity, QuantityValue, & Unit).

We have populated two online recipes to RecipeOn ontology as shown in Figure. 4.18 and Figure. 4.19. There was a complete overlapping between the recipe classes and the classes mentioned in the ontology.

4.3.2.2 Adaptability

We focused on adaptability in two aspects. Firstly, we designed RecipeOn with the aim of reusability so that future developed related ontologies may extend or use components from RecipeOn ontology. We have reused concepts and properties from recipe schema and existing ontology design patterns(including owl, seq, qudt, rdf, and xsd) to encourage adaptability. Also, we have added class

[23]https://oops.linkeddata.es/
[24]https://www.similarweb.com/top-websites/category/food-and-drink/cooking-and-recipes/
[25]https://schema.org/Recipe
[26]http://ontologydesignpatterns.org/cp/owl/sequence.owl
[27]http://qudt.org/2.1/schema/qudt

labels and class descriptions to increase adaptability. Secondly, we generated documentation[28] of RecipeOn ontology using LODE[29]. This would help to increase RecipeOn's ontology understanding and hence adaptability.

4.3.2.3 Clarity

To avoid ambiguity and increase human understandability of concepts and properties, we have added labels and comments where required. This not only increases the clarity of concepts and properties but also makes them more expressive.

4.3.2.4 Completeness

To evaluate the completeness of RecipeOn ontology, we have designed 28 competency questions. Based on the similarity these competencies have been grouped into five modules(i.e. core-recipe, ingredients, actions, procedures, and nutrition). Each module covers a dimension of the recipe process. We have written SPARQL queries against each competency question as mentioned in Table 4.3, Table 4.4, Table 4.5, Table 4.6, and Table 4.7. RecipeOn fulfills the scope and answers all competency questions as mentioned in Table 4.1.

4.3.2.5 Consistency

We found a few inconsistencies in RecipeOn ontology when reasoning was performed using HermiT 1.4.3.456[30]. RecipeOn is made consistent and all the inconsistencies have been removed.

4.3.3 Ontology Metrics

The detailed ontology metrics for RecipeOn are presented in Table 4.8 as a summary of total classes, data properties, object properties, and axioms.

[28]https://hajirajabeen.github.io/EvoRecipesOntology/
[29]Live OWL Documentation Environment https://essepuntato.it/lode/
[30]http://www.hermit-reasoner.com/

TABLE 4.8: Ontology Metrics.

Metrics	
Axiom	1387
Logical axiom count	492
Declaration axioms count	346
Class count	231
Object property count	37
Data property count	20
Individual count	52
Annotation Property count	16
Class Axioms	
SubClassOf	209
EquivalentClasses	55
DisjointClasses	8
GCI count	0
Hidden GCI Count	54
Object Property Axioms	
SubObjectPropertyOf	8
EquivalentObjectProperties	0
InverseObjectProperties	5
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrrefexiveObjectProperty	0
ObjectPropertyDomain	37
ObjectPropertyRange	37
SubPropertyChainOf	2
Data Property Axioms	
SubDataPropertyOf	0
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	20
DataPropertyRange	20
Individual Axioms	
ClassAssertion	0
ObjectPropertyAssertion	79
DataPropertyAssertion	12
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0
Annotation Axioms	
AnnotationAssertion	895
AnnotationPropertyDomain	0
AnnotationPropertyRangeOf	0

4.4 RecipeOn Example

For many people, cooking is difficult and without a proper recipe, it can become more challenging. *RecipeOn* ontology helps users to understand the recipe process as it not only presents the ingredients and actions but also the sequence of actions and the ingredients used in each of the actions. To elaborate on this idea, we have mapped "Roast Beef Dip Sandwich with Herbed Garlic Au Jus"[31] recipe using *RecipeOn* and depict its knowledge graph as shown in Figure 4.18. Red beef and garlic herb boursin cheese are the main ingredients. Soy sauce, thyme, red wine, basil, Worcester shire sauce, onion, and garlic are the flavor ingredients while beef broth, french baguette, and olive oil are the side ingredients. There are 14 actions that are used in this example including heat, add, cook, stir, simmer, spread, divide, refrigerate, and bake. It is clear from the figure that actions follow a sequence (e.g. Add follows Heat, Cook follows Add, and Stir follows Cook). Moreover:hasIngredient property represents ingredients that are used in each action (e.g. Heat action uses olive oil and spread action uses garlic herb boursin cheese). It is clear from the figure that all actions are initiated after the ingredients and actions follow a sequence using `directlyPrecedes` relationship.

4.5 New Recipe Generation: The Use Case of RecipeOn

Generating novel recipes is a challenging task and has gained much attention in recent years. It is a natural curiosity that we want to try new recipes and for this reason we synthesize the recipes by altering the ingredients or actions. The goal of this alteration could be different for different types of intended users. For example, pre-diabetic (diet-conscious) persons avoid a high calorie diet and they opt for low calorie alternative ingredients. Similarly, heart patients are advised to avoid some particular action (such as fried) recipes. Food enthusiasts may not have few ingredients at home and will look for potential recipes that can be created using the ingredients at hand. Keeping in view the needs of the intended users, we have formulated a few competency questions related to new recipe generation.

A Can we **bake** patties instead of **shallow frying** (for heart patients)?

[31] https://www.food.com/recipe/roast-beef-dip-sandwich-with-herbed-garlic-au-jus-352204

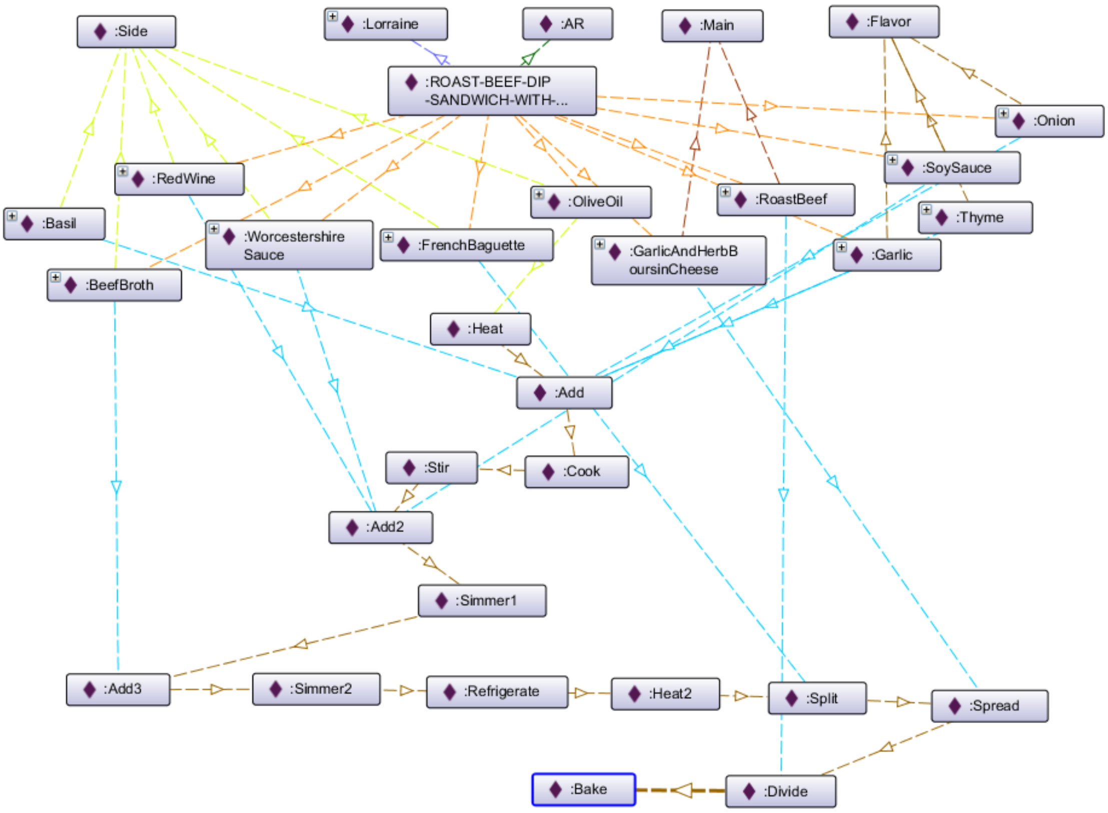

FIGURE 4.18: Knowledge graph of "Roast Beef Dip Sandwich with Herbed Garlic Au Jus" Recipe

B Can we **spray olive oil** on top of chicken pasta rather than **layering cheese**?

C Can we exchange **chicken** with **mutton** to increase the protein content in pasta?

D Can we replace **white sugar** with **brown sugar** in cold coffee?

E Can I use **vinegar** instead of **lemon juice** (due to unavailability of lemons in current season)?

F Can I **shred potatoes** instead of **cutting** while preparing french fries?

G Can I use **boiled noodles** instead of boiled rice in **oven**?

In the following subsections, we discuss how RecipeOn supports the new recipe generation process using alternative ingredients and actions.

4.5.1 Alternative Ingredient/Action

Alternative ingredients/actions can be compared to the mutation operation of evolutionary algorithms in which we do minor changes in an individual (recipe) to generate a new mutated individual (novel recipe). `alternateAction` and `hasAlternateIngredient` properties are used to replace `Ingredient` and `action` with their substitutes. In a comparison of ontological and non-ontological resources (i.e, Thai Food Ontology for Ingredient Substitution (Saengsupawat et al., 2017), Ontology-based Knowledge Acquisition for Thai Ingredient Substitution (Saengsupawat et al., 2014), Searching cooking recipes by focusing on common ingredients (Kusu et al., 2017), Flavor Graph (Park et al., 2021), Modification of food recipes based on the geographic origin of produce (Lu et al., 2020), SmartChef (Draschner et al., 2019), and Veganaizer (Lawo et al., 2020)) we found AutoChef (Jabeen et al., 2020) more suitable approach for RecipeOn to find ingredient and action substitutes. AutoChef uses Ingredient-Ingredient and Action-Action replacement mechanisms as shown in Table 4.9 and Table 4.10 respectively. An ingredient i can replace ingredient j if and only if ingredient i is in the alternative list of ingredient j (For example Turnip can replace Sweet Potato according to Table 4.9). Similar is the case with Actions (For example, cut and Dice can replace, according to Table 4.10).

TABLE 4.9: Ingredients and Alternative Ingredients

Sr.	Ingredient	Alternative Ingredient
1	Asparagus	Broccoli, Okra, Snap-Beans
2	Beets	Cucumbers, Zucchini
3	Broccoli	Okra, Snap-Beans, Asparagus
4	Cucumbers	Zucchini, Beets
5	Okra	Broccoli, Snap-Beans, Asparagus
6	Potatoes	Sweet Cassava, Sweet Potatoes, Turnip
7	Snap-Beans	Broccoli, Okra, Asparagus
8	Sweet Cassava	Sweet Potatoes, Potatoes, Turnip
9	Sweet Potatoes	Sweet Cassava, Potatoes, Turnip
10	Turnip	Sweet Potatoes, Sweet Cassava, Potatoes
11	Zucchini	Cucumbers, Beets

4.5.2 Alternative Procedure

An alternative procedure is analogous to the crossover operation of evolutionary algorithms. There is no ontological resource available for the alternative procedure. This module is in comparison to the concept of "Ingredients with state" as mentioned in EvoChef (Jabeen et al., 2019). A procedure is a set of

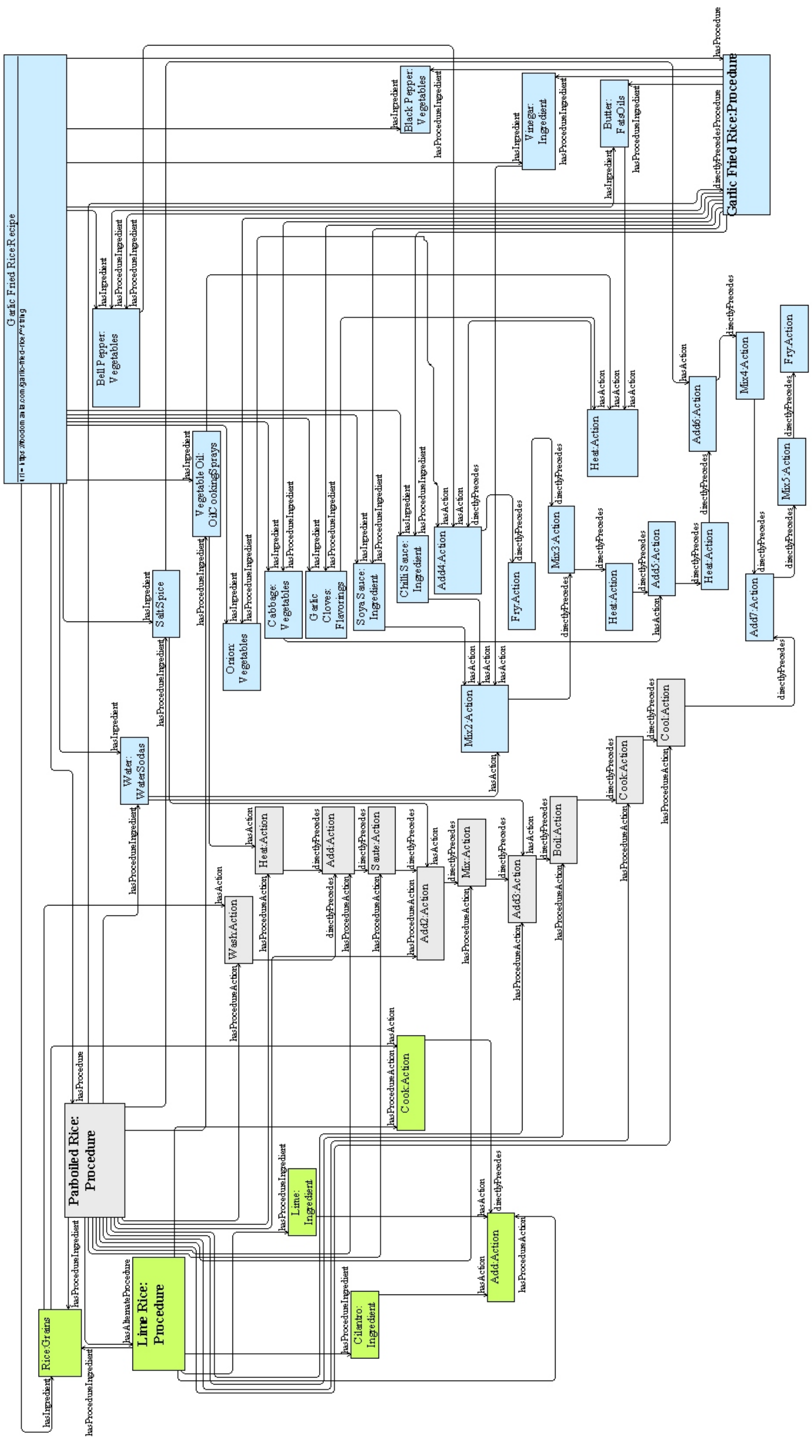

FIGURE 4.19: Rice Recipe with Alternative Procedure

TABLE 4.10: Actions and Alternative Actions

Sr.	Action	Alternative Action
1	Blend	Stir, Mix
2	Boil	Steam
3	Chill	Cool
4	Chop	Dice, Cut
6	Cool	Chill
7	Cut	Dice, Chop
8	Dice	Cut, Chop
9	Dip	Soak
10	Mix	Stir, Blend
11	Rinse	Wash
12	Soak	Dip
13	Steam	Cucumbers, Beets
14	Stir	Boil
15	Wash	Clean, Rinse

processed ingredients or in other words, ingredients with corresponding actions. `hasAlternateProcedure` object property is used to substitute a procedure with another procedure. To understand the concept we have mapped "Garlic Fried Rice"[32] recipe on RecipeOn and then used the property `hasAlternateProcedure` to substitute a procedure. "Garlic Fried Rice" recipe has two procedures i.e, Parboiled rice and garlic fried rice. Actions and Ingredients involved in parboiled rice and garlic fried rice procedures are represented with gray and blue colors as shown in Figure 4.19. Lime rice procedure is part of another recipe (i.e, "Camarones a la Diabla"[33]). This procedure with its ingredients and actions has been shown with green color in Figure 4.19. Lime rice procedure substitutes parboiled rice and all green colored nodes replace gray colored nodes.

4.6 Summary

In this chapter, we present RecipeOn, an Ontology for representing cooking Recipes. It follows a modular approach that comprises core-recipe, ingredients, actions, nutritions, and procedure modules. RecipeOn successfully maps

[32] https://foodomania.com/garlic-fried-rice/
[33] https://www.acozykitchen.com/camarones-a-la-diabla

any given recipe as a process and is capable to answer the competency questions designed for intended users (sick persons, diet-conscious people, and food enthusiasts). RecipeOn is designed following NeOn methodology for ontology development while using the classes and properties from existing ontologies (i.e., owl, Seq, schema, and Qudt). Moreover, RecipeOn can be applied to our specifically designed AI-based use case of novel recipe generation, achievable through alternative ingredients, instruction, and procedures. RecipeOn is also consistent with Ontology quality parameters. RecipeOn and its documentation is available online and hence it is designed to be adaptable and reusable.

Chapter 5

RecipeKG: Knowledge Graph to Manifest Culinary Recipes

Creativity is a complex concept that is hard to define and difficult to evaluate. It not only refers to creating something new but may also relate to solving old problems in a new way. Alongside the creation of new ideas, creativity also focuses on the application of those ideas. Creativity adds value to the solutions by improving their quality and acts like a force that compels evolution and change. Scientists modeled the concept of creativity to computer science using different techniques and algorithms to simulate human creativity and named it computational creativity (Ławrynowicz, 2020). Computational creativity is based on concepts from psychology, arts, and philosophy. Computational creativity is used to generate new ideas: (i) By combining existing solutions or their parts to create new solutions that did not exist; (ii) By transforming the solutions to generate novel solutions that were difficult to be created using the traditional creativity process; (iii) By exploring high dimensional conceptual spaces that are difficult to be explored using traditional creativity process. Computational creativity doesn't assure that computers are creative but demonstrates that computers can model and simulate human creativity to generate novel ideas/-solutions. Linguistic creativity, music creativity, visual art creativity, and creative problem-solving are the few success stories of computational creativity (Loughran and O'Neill, 2017).

Cooking is an interesting case study to explore computational creativity. Cooking is both an art and a science. It's an art as there is too much subjectivity involved in terms of taste, color, texture, aroma, and style. It's a science as it involves chemical components present in ingredients and many chemical processes, to transform ingredients into edible food. Creativity in culinary recipes can be explored in multiple contexts: (i) Changing ingredients or a combination of ingredients to create a novel recipe; (ii) Exploring multi-dimensional ingredients search space to find a potential ingredient combination that could generate a novel recipe. Humans generally do this using intuition as the human mind is not capable to explore large search space; (iii) Applying new actions on existing ingredients that may change the aroma; (iv) Replacing the sub-process (that involves both ingredients and actions) of a recipe with an alternate potential sub-process.

There are several recipe datasets published in the last decade by considering different objectives, we have summarized them in section 5.1. The main motivation behind building a detailed recipe dataset is to emphasize and highlight different components involved in the recipes. Our focus is to highlight different interesting patterns like the most frequently occurring ingredient in recipes and their correlation with other frequent ingredients, correlation among ingredients, and actions. Furthermore, we proposed the $RecipeKG$ (Recipe Knowledge Graph) on the top of the 0.8 million recipe dataset to act as a source for the computational creativity of recipes.

The contribution of this chapter is as follows.

- We have created a $0.8M - Recipes$ recipe dataset that comprises 810000 recipes where each recipe is represented using a plurality of 50+ attribute values.
- We have generated $RecipeKG$ on top of $0.8M - RecipeKG$ comprises 209 million facts and represents each recipe as a process. Process representation helps to create machine-generated recipes using appropriate alternative actions and ingredients.
- $RecipeKG$ provides ingredient-ingredient, ingredient-action, and action-action relationships using the concepts and properties from $RecipeOn$ ontology.

This chapter will cover the following research question.

RQ3. Is there any recipe dataset with a plurality of attributes that can be used to create a recipe knowledge graph?

This chapter is based on the following article.

> ***Muhammad Saad Razzaq***, *Fahad Maqbool, Hajira Jabeen, RecipeKG: Knowledge Graph to Manifest Culinary Recipes", to be submitted in Knowledge-Based Systems.*

The rest of the chapter is organized as follows, 0.8 Million Recipes dataset is discussed in section 5.1, it contains details about data extraction, data consolidation, data preprocessing, data transformation, Ingredient transformation, Instruction transformation, and dataset entities. The recipe knowledge graph has been discussed in section 5.2. Use cases and discussion is illustrated in section 5.3.

5.1 0.8 Million Recipes (0.8M-Recipes)

Culinary cooking is a complex task dependent on many factors (selection of ingredients, proportions of ingredients, the substitution of ingredients, actions on ingredients, and sequence of actions. Food experts say that the simplest recipes (e.g. boiled rice, baked chicken, french fries) are often difficult to cook as there is less number of choices to hide behind fancy items (i.e. sauces, sprinkles, garnish). This further increases the significance of the above-mentioned factors. Due to the complexity of the cooking process, learning culinary recipes requires a large-scale collection of recipes (like Recipe1M+[1] (Marin et al., 2019)). In recent years, the volume of the recipe datasets has increased from a few thousands (Bossard et al., 2014, Chen et al., 2009, Jain, 2020) to million plus items (Harashima et al., 2016, 2017, Salvador et al., 2017). However, less attention has been paid to rich recipe features (i.e. relevant data and metadata) extraction. Understanding culinary recipes demands rich metadata. This may include descriptive metadata (like published date, author, cooking time, title, etc) or contextual metadata (utensils used during an action).

Numerous applications of the food domain demand exploration of food in a multitude of dimensions (i.e. recipe representation (Li and Zaki, 2020, Pallagani et al., 2022), recipe generation (Agarwal et al., 2009, Antô et al., 2020, Draschner et al., 2019, Jabeen et al., 2019, 2020, Naik and Polamreddi, 2015, Varshney et al., 2019), food creativity & assessment (Engisch, 2020, Jimenez-Mavillard and Suarez, 2022, Pinel and Varshney, 2014, Sakib et al., 2022), food management (Agarwal, 2020), ingredient substitution (Gim et al., 2022, Lu et al., 2020, Nadee and Unankard, 2021, Pan et al., 2020, Shirai et al., 2017), food recommendation (Beijbom et al., 2015, Elsweiler et al., 2017, Gim et al., 2021, Khilji et al., 2021, Lei et al., 2021), image to food retrieval (Bolaños et al., 2017, Chen and Ngo, 2016, Chen et al., 2020, 2018), food-based resource creation (Popovski et al., 2019), and food matching (Arffa et al., 2016, Su et al., 2014). RecipeDB[1] (Batra et al., 2020) has been developed to facilitate basic and advanced search based on ingredients, cuisines, flavor, plus along with 1M+ user interactions. We have compiled the statistics for some of the famous datasets as given in table 3.1 based on volume, number of features, nutritional information, and data format. In this chapter, we have developed 0.8M-Recipes dataset [15] that

- contains around 0.8M-Recipes dataset.
- represents each recipe using a rich feature set of 50+ attributes.
- is available in SQL format.

5.1.1 Data Extraction

We have analyzed several recipe datasets to act as a source of information/-knowledge for *RecipeKG* but none of them was fully capable to fulfill all the requirements of *RecipeKG*, for example, Recipe1M+ (Marin et al., 2019) lacks cuisine, course, timings, and nutritional information; RecipeNLG (Bień et al., 2020) lacks the data about nutrition, recipe image, and user interactions; RecipeDB (Batra et al., 2020) and food.com (Majumder et al., 2019) relatively cover many attributes but they lack the heterogeneity of data and have less

[1]https://cosylab.iiitd.edu.in/recipedb

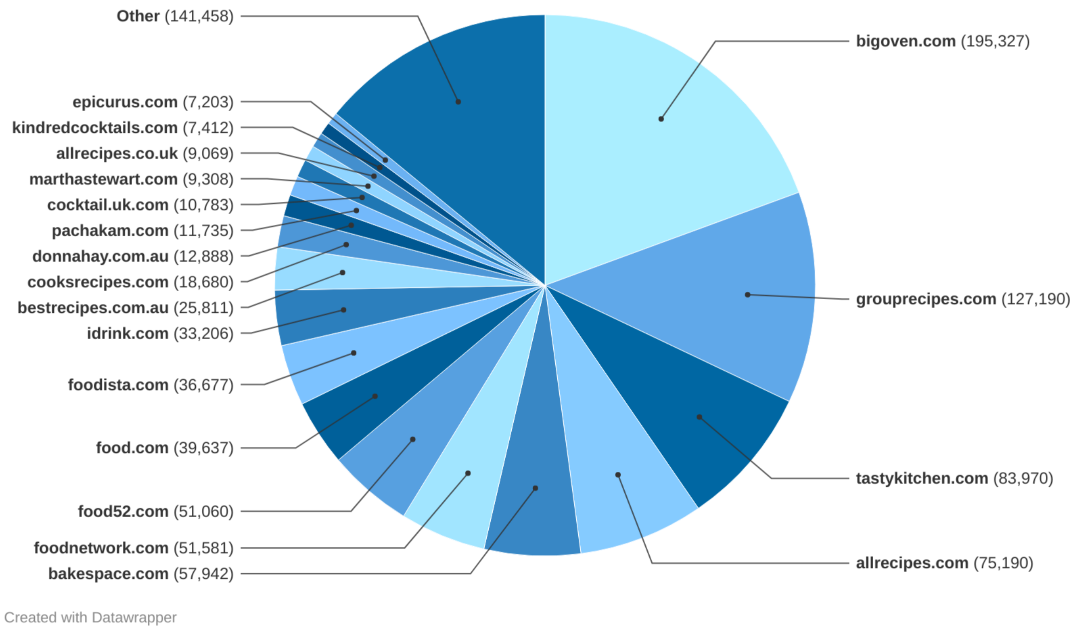

FIGURE 5.1: Recipes Extracted from Top 50 Recipe Websites

volume compared to the counterparts. Therefore we selected top 500 (previously based on alexa[2] rank, now similarweb[3] score) recipe websites for data extraction. Data is extracted using Visual Web Ripper (VWR) [4]. Extracted data contains four main entities that are

- *Recipe:* It contains 0.8 million recipes with 30+ attributes. The main attributes include recipe name, URL, recipe category, cuisine, course, recipe complexity, and recipe image.
- *Ingredient:* It contains 8.95 million ingredients against 0.8 million recipes. Ingredient quantity, unit, and ingredient name are the main attributes of the ingredient entity.
- *Instruction:* It contains 6 million+ instructions. Sequence, instruction, action, and ingredient are the main attributes of the instruction entity.
- *Nutrition:* it contains 0.5 million records and 39 attributes. The main attributes are iron, fiber, carbohydrates, cholesterol, fat, and vitamin C.

A separate script file has been generated for data extraction from each recipe website and attributes were extracted that are compatible with schema.org[5].

[2] https://www.alexa.com/
[3] https://www.similarweb.com/
[4] http://www.visualwebripper.com, Accessed Oct 02, 2021
[5] https://schema.org/Recipe

Separate script files have been created due to the variable structure and layout of source websites. A number of recipes extracted from different websites are shown in figure 5.1. Data is placed at GitHub repository[15].

5.1.2 Data Consolidation

The richness of recipe data/metadata may differ on different recipe websites due to variable amount of available information (e.g. "Oreo no-bake cookies"[6] recipe from allrecipes.com has nutritional information which is missing in "smoked prime rib for Christmas"[7] recipe from smoking-meat.com). Attributes have different formats on different websites. Different field names of the same property add another challenge in data consolidation. For an organized consolidation of data we created a new database in MySQL with four entities (i.e. Recipes, Ingredients, Instructions, and Nutrition) as shown in figure 5.8. Attribute names in RecipeDB are standardized based on Recipe Schema[5]. We have discussed these entities in 10 digits unique id where the first 4 digit represents the Alexa rank of the website while the last 6 digits represent the recipe's unique identification.

5.1.3 Data Preprocessing

Data cleaning/preprocessing improves the quality of data before we may apply transformation techniques to map extracted data to relevant attributes. It involves following steps.

- a Remove complete statements.
- b Remove partial statements.
- c Conversion of symbols and special characters.
- d Conversion of units.

Removal of complete statements includes optional elements or special notes in ingredient phrases/instruction statements that start with specific keywords

[6] https://www.allrecipes.com/recipe/276669/oreo-no-bake-cookies/
[7] https://www.smoking-meat.com/december-22-2015-smoked-prime-rib-for-christmas

("note" or "optional"). Table 5.1 shows a few data cleaning operations with examples.

Sr.	Before Preprocessing	After preprocessing	Operations
1	25 g unsalted nuts, (whatever kind you have)	25 g unsalted nuts	Remove text in brackets
2	3-4 sprigs of woody herbs such as rosemary, thyme, flat-leaf parsley	3-4 sprigs of woody herbs	Remove text after "such as"
3	100 g rustic bread, torn into 2.5cm pieces	100 g rustic bread	Remove text after period
4	Hunt's® Bruschetta Chicken Skillet	Hunt's Bruschetta Chicken Skillet	Remove special symbols like © ™ ®
5	1 large handful of dried cranberries or raisins	1 large handful of dried cranberries	Remove textt after or clause
6	*Note: you can use 2 -3 tsp vanilla extract	-	Remove any note statements.
7	OPTIONAL: Chocolate sauce	-	Remove optional items
8	edible flowers , to garnish	edible flowers	Remove text after comma
9	1. Put the four lime wedges into a glass.	Put the four lime wedges into a glass.	Remove serial numbers
10	2677 kcal	2677	Remove units in nutrient fields
11	1 hour 5 minutes	65	Time converted to minutes for prepTime, cookTime, & totalTime
12	63 comments	63	Removed unnecessary text from commentCount
13	Transfer the jalapeénos to a bowl to cool	Transfer the jalapenos to a bowl to cool	Convert Latin letters like ã, é, õ to English letters.
14	½ pound smoked sausage	1/2 pound smoked sausage	Convert ascii characters to alpha numeric text

TABLE 5.1: Sample Data from 0.8M-Recipes and Corresponding Operations for Data Cleansing

5.1.4 Data Transformation

Data Transformation is a two-step process with the aim to extract: (i) Ingredients from ingredient phrases; (ii) Action and ingredient pairs from instruction statements. Both ingredient transformation and instruction transformation has been discussed in detail in the following sub-sections.

5.1.4.1 Ingredients Transformation

Transformation of ingredient phrases to relevant attributes (i.e. quantity, unit, size, ingredient) was a challenging task due to the lexical variation of ingredient phrases. Previous studies (Artese et al., 2021, Ozgen, 2019, Saenz-Gardea, 2019, Silva et al., 2019) suggested POS tagging to identify attributes from ingredient phrases using popular NLTK (Bird, 2006) [8], Spacy [9], and Stanford Core NLP (Manning et al., 2014) [10] libraries. These studies suggested that Stanford Core NLP and Spacy perform better than NLTK (Schmitt et al., 2019). Therefore, we shortlisted these two libraries to do a performance comparison on a sample of 100 ingredient phrases.

2	x	400	g	tins	cannelini	beans
NUM	SYM	NUM	NOUN	NOUN	VERB	NOUN
Qnt		---	Un	Ing		Ing

25	g	unsalted	nuts
NUM	DET	ADJ	NOUN
Qnt	---		Ing

FIGURE 5.2: POS Tagging of Ingredient Phrases Using the Spacy Library. In the First Ingredient Phrase "400 g" is the Size and "tins" is the Unit. These both have been Misidentified. In the second phrase "g" has also been missed which should be Mapped to Unit.

[8] https://www.nltk.org/
[9] https://spacy.io/
[10] https://stanfordnlp.github.io/CoreNLP/

2	x	400	g	tins	cannelini	beans
CD	NN	CD	NN	NNS	NN	NNS
Qnt	Un		Ing	Ing	Ing	Ing

25	g	unsalted	nuts
NUM	DET	ADJ	NOUN
Qnt	---		Ing

FIGURE 5.3: POS Tagging of Ingredient Phrases using Stanford Core-Nlp Library. In the first Ingredient Phrase "x" has been Misidentified as a unit. In the Second Phrase "g" has been Missed.

2	x	400	g	tins	cannelini	beans
Qnt		Size	Size	Un	Ing	Ing

25	g	unsalted	nuts
Qnt	Un	Ing	Ing

FIGURE 5.4: Identification of Quantity, Unit, Size, and Ingredient using Regular Expressions. In both Ingredient Phrases Attributes have been Correctly Identified.

Identification of phrases using both libraries has issues (i.e. mis identification of attributes or nonidentification of attributes) as shown in figure 5.3 and figure 5.4. However, our designed regular expressions have correctly identified all the attributes in both ingredient phrases as shown in figure 5.4. Moreover, we have performed a performance comparison of attributes identification using Stanford core-NLP, spacy, and regular expressions on a sample of 100 ingredient phrases. Attributes identification using regular expressions achieve better performance as shown in table 5.2.

5.1.4.2 Instructions Transformation

Long text of instruction statements creates another challenge in the extraction of actions and relevant ingredients from instruction statements. We also used Spacy and Stanford core-NLP libraries to identify ingredients and actions from instruction statements as shown in figure 5.5 and figure 5.6 respectively. Furthermore, we designed regular expressions for the identification of actions and string matching for the identification of ingredients from instruction statements as shown in figure 5.7.

TABLE 5.2: Statistics Summary of Ingredient Phrases Transformation using Stanford Core-NLP, Spacy, and Regular Expressions.

Metrics	Core-NLP	Spacy	Reg Expression
False Positive	30	35	**12**
True Positive	85	87	**108**
False Negative	30	27	**07**
True Negative	95	91	**113**
Accuracy	0.75	0.742	**0.921**
F1	0.739	0.737	**0.919**
Recall	0.739	0.763	**0.939**
Selectivity	0.76	0.722	**0.904**
Precision	0.739	0.713	**0.9**
Negative Predictive Value	0.76	0.771	**0.942**
Miss Rate	0.261	0.237	**0.061**
Fall Out	**0.24**	0.278	0.096
Fall Discovery Rate	0.261	0.287	**0.1**
False Omission Rate	0.24	0.229	**0.058**

To	make	the	crispy	onions		**peel**	and	thinly	**slice**	the	**shallots**	or	**onion**	into	rings	.
PART	**VERB**	DET	**NOUN**	**NOUN**	PUNCT	**NOUN**	CCONJ	ADV	**VERB**	DET	**NOUN**	CCONJ	**NOUN**	ADP	**NOUN**	PUNCT.
	Act		**Ing**	**Ing**		**Ing**			**Act**		**Ing**		**Ing**		**Ing**	

FIGURE 5.5: POS Tagging of Instruction Statements using the Spacy Library.

To	make	the	crispy	onions	,	**peel**	and	thinly	**slice**	the	**shallots**	or	**onion**	into	rings	.
TO	**VB**	DT	JJ	**NNS**	,	**NN**	CC	RB	**VB**	DT	**NNS**	CC	**NN**	IN	**NNS**	.
	Act			**Ing**		**Ing**			**Act**		**Ing**		**Ing**		**Ing**	

FIGURE 5.6: POS Tagging of Instruction Statements using Stanford Core-NLP Library.

To	make	the	crispy	onions,	**peel**	and	thinly	**slice**	the	**shallots**	or	**onion**	into	rings.
					Act			**Act**		**Ing**		**???**		

FIGURE 5.7: Ingredients and Actions Extraction from Instruction Statement using Hybrid Approach

A sample of 100 randomly selected instruction statements was selected to do a performance comparison of Spacy, Stanford core-NLP, and hybrid approach (regular expression + string matching). Table 5.3 show that the hybrid approach achieves better performance compared to Spacy and Stanford core-NLP.

5.1.5 Dataset Entities

0.8M-Recipes is a MySQL database comprising of four main entities (i.e. CoreRecipe, Ingredients, Instructions, and Nutrition) as shown in figure 5.8. CoreRecipe stores the basic data, metadata, and provenance information related

Table 5.3: Statistics Summary of Instruction Statements Transformation using Stanford Core-NLP, Spacy, and Hybrid Approach.

Metrics	Core-NLP	Spacy	hybrid approach
False Positive	85	73	**03**
True Positive	44	**45**	43
False Negative	**02**	03	06
True Negative	173	183	**252**
Accuracy	0.714	0.75	**0.97**
F1	0.503	0.542	**0.905**
Recall	**0.957**	0.938	0.878
Selectivity	0.671	0.715	**0.988**
Precision	0.341	0.381	**0.935**
Negative Predictive Value	**0.989**	0.984	0.977
Miss Rate	**0.043**	0.063	0.122
Fall Out	0.329	0.285	**0.012**
False Discovery Rate	0.659	0.619	**0.065**
False Omission Rate	**0.011**	0.016	0.023

to each recipe. Ingredients store a list of ingredients against each recipe and Instructions store a list of directions to prepare and cook food. Finally, Nutrition stores the available nutrient information against each recipe. Each of these entities has been discussed in detail in the following sub-sections.

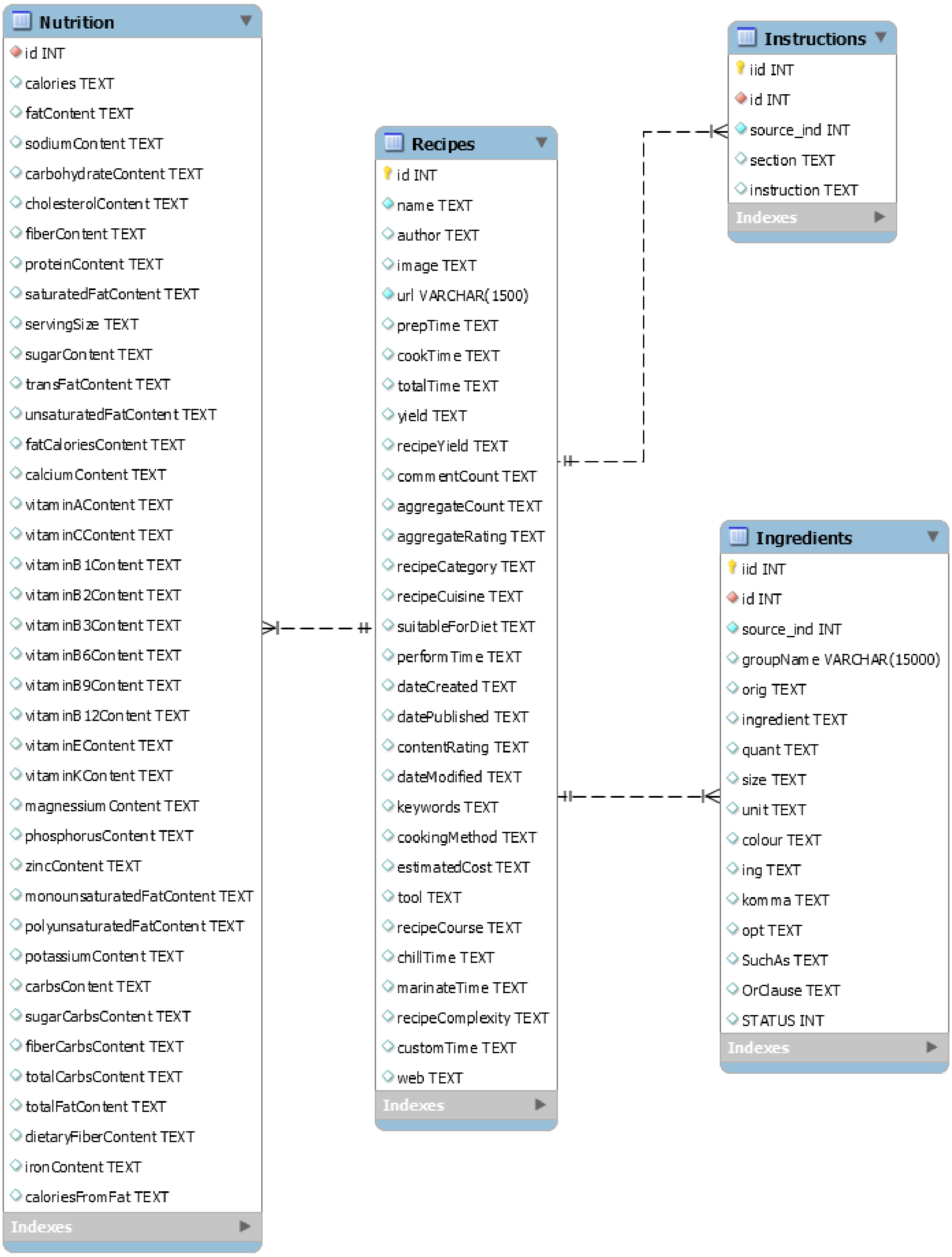

FIGURE 5.8: CoreRecipe has 1-M Relationship with Ingredients and Instructions, while 1-1 Relationship with Nutrition.

5.1.5.1 Entity: CoreRecipe

CoreRecipe is the central entity of 0.8M-Recipes that comprises basic recipe features (like id, name, image, category, course, cuisine), recipe metadata (like yield, preparation time, cooking time, total time, comment count, aggregate count, aggregate rating), and provenance information (like URL, author, date-Published, date-Updated). *CoreRecipe* has 810,000 recipes. Each recipe is rep-

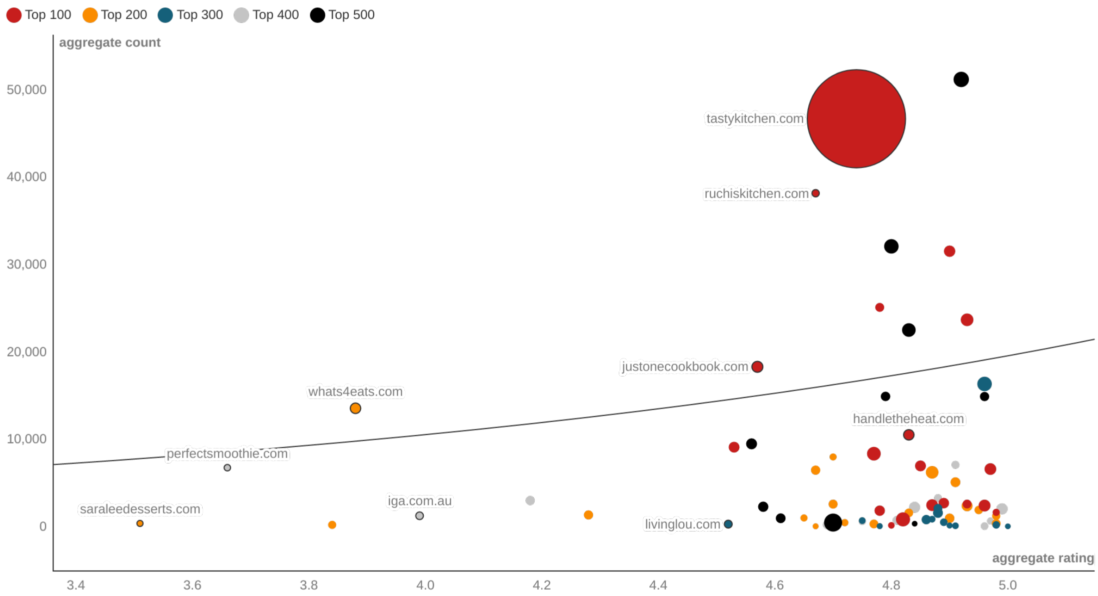

FIGURE 5.9: Average Aggregate Rating of each Recipe Website Against the Total Number of Aggregate Ratings. The Color of the Sphere Shows the Ranking of the Website Based on Traffic Volume while the Size of the Sphere Shows the Volume of the Recipes on the Website.

resented using a 10-digit unique id and a unique URL categorized among 81 cuisine styles and 271 recipe categories. Moreover, these recipes also mention a set of 83 cooking tools used in the cooking process. Aggregate rating and total ratings against each recipe are also part of the *CoreRecipe*. Figure 5.9 shows the average rating of the recipes of each website against the total recipe ratings of the website. "Tasty Kitchen"[11] is the highest recipe volume website that has recipes with high average ratings. On the other hand, "Sara Lee Desserts"[12] has recipes with the worst average rating.

5.1.5.2 Entity: Ingredients

Ingredients play a vital role in a recipe due to their nutritional, functional, and sensory characteristics. They add color, taste, texture, and aroma to the food. Ingredients also have a key role in the ingredient substitution process while

[11] https://tastykitchen.com/
[12] https://saraleedesserts.com/

generating novel recipes. Ingredients exist in recipes both in raw form (such as potatoes, rice, or tomatoes) and pre-processed form (baked potatoes, boiled rice, mashed potatoes).

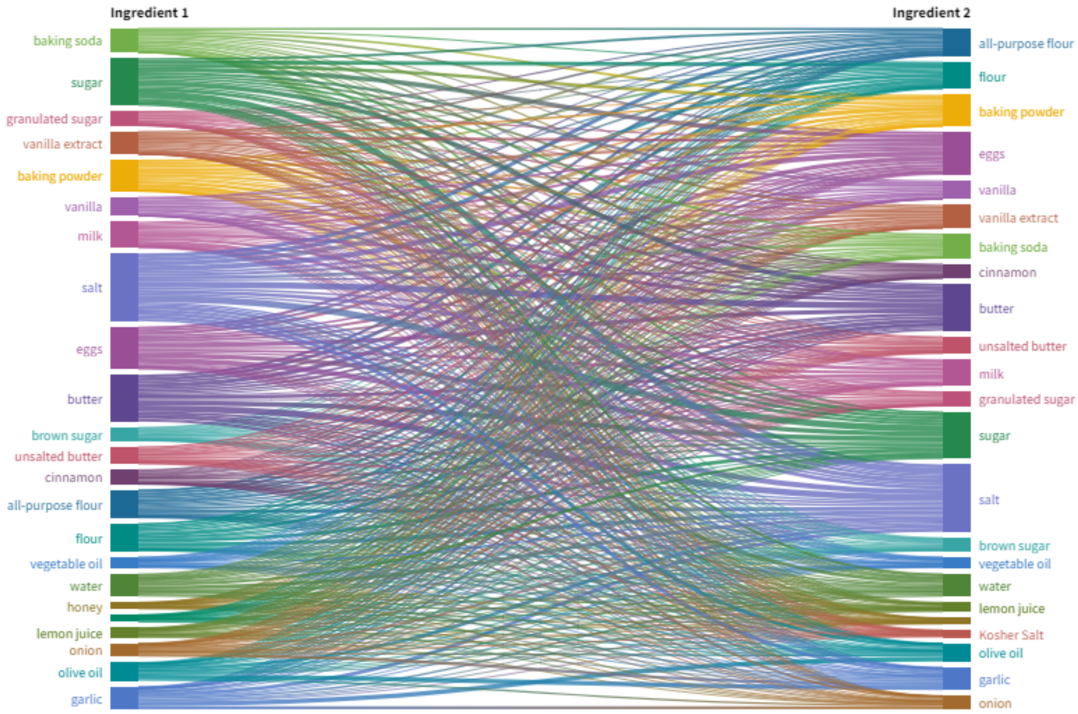

FIGURE 5.10: Bipartite Graph representing (most frequently occurring) Ingredients as Nodes having a Relationship with other (most frequently occurring) Ingredients based on Co-Existence in the same Recipe.

Ingredients entity comprises a total of 8.95 million ingredients while 0.61 million unique ingredients. Each ingredient is identified using a unique ingredient id (i.e. *iid*). 72185 recipes group the ingredients based on their role in different procedures (e.g. "Sweet Potato Shepherd's Pie"[13] groups the ingredients in two procedures (i.e. (i) Filling & (ii) Sweet Potato Topping)). There are 910182 ingredients that have been grouped into 51512 distinct procedures. Quantity, unit, and size information are also available with ingredients (e.g. "2 x 400 g tins cannellini beans" has quantity 2, size 400g, unit tin, and ingredient cannellini beans). There are 7 million ingredients that have 50+ distinct units of information attached to them.

Ingredient extraction has imposed many challenges during the extraction and attribute standardization phase. These challenges include:

[13] https://www.acozykitchen.com/sweet-potato-shepherds-pie-whole30

Lexical structure of ingredient phrases Ingredient phrases can be precise (like "olive oil", "sea salt"), simple (like "50 g pine nuts", "1 teaspoon runny honey"), complicated (like "2 x 125 g balls of mozzarella cheese", "4 large raw peeled Tiger prawns"), or complex (like "½ a bunch each of Chinese chives, Thai basil, Thai mint, (45g total)", "3 good handfuls mixed fresh herbs (mint, parsley, oregano, thyme), leaves picked"). The variation in a lexical structure created a challenge in classifying the components of ingredient phrases using POS tagging as mentioned in section 5.1.4.

Attributes extraction from ingredient phrases Quantity, unit, size, and ingredient appear in variable sequences in ingredient phrases. Ingredient phrases may start with quantity following unit (like "4 tablespoons extra virgin olive oil", "150 ml semi-skimmed milk"), quantity following size (like "3 long aubergines", "2 x 200 g tuna steaks"), quantity following ingredient (like "3 banana shallots", "2 red shallots"), ingredient following quantity (like "Onion - 2", "Red chilly powder - 2 tbsp"), and many other variations. Moreover, ingredient phrases may not contain one or more attributes (like "Herb pesto" only has an ingredient, "1 lemon" has only quantity and ingredient, "Ginger: small piece" does not has quantity). The sequence variation makes it challenging to extract the attributes (i.e. quantity, unit, size, & ingredient) by considering the existence of an attribute relative to the existence of another attribute (e.g. its not necessary that unit will always appear after the quantity or ingredient will follow the unit).

Examples, notes, & alternative/optional ingredients Ingredient phrases contain example statements that are usually mentioned using the keyword "Such As" (e.g. "3-4 sprigs of woody herbs, such as rosemary, thyme, flat-leaf parsley"), note statements mentioned using "*" (e.g. "*Salt and Black Pepper to taste"), alternative ingredients mentioned using the keyword "or" (e.g. "3 tablespoons marsala or sweet sherry", "1 cinnamon stick, or 1 pinch of ground cinnamon"), and optional ingredients (e.g. "sesame oil, optional", "optional: chocolate sauce"). These keywords (i.e. such as note, or, optional) add extra information to ingredient phrases that are not needed in the attribute extraction phase and hence were eliminated during data cleansing.

5.1.5.3 Entity: Instructions

Instructions are an essential part of a recipe and minimally recipe information includes a list of ingredients and a set of instructions. Instructions represented on recipe websites are presented using a sequential structure. The sequence is usually expressed using an alphanumeric list or simply bullets. Each instruction statement may comprise multiple instruction sentences where each sentence comprises a set of actions and a subset of ingredients on which these sets of actions are performed.

Instructions although follow some chronological order and seem a sequential set of activities. But in practice, actions a_i and a_j using ingredients $ingS_i$ and $ingS_j$ respectively can be performed in parallel when an action a_i does not need the same raw or processed ingredients from another action a_j (i.e. $ingS_i \cap ingS_j = \varnothing$) and hence an action a_i do not directly follows another action a_j. To find actions and ingredients dependency from given instructions, it was necessary to first extract the ingredients and actions from instructions. To make the extraction simple we segmented each instruction (ins_p) to one or more individual sentences($s_p^1 \cdots s_p^q$) based on the period (.) symbol. As a result, 5.14 million instructions were segmented into 10.89 million sentences.

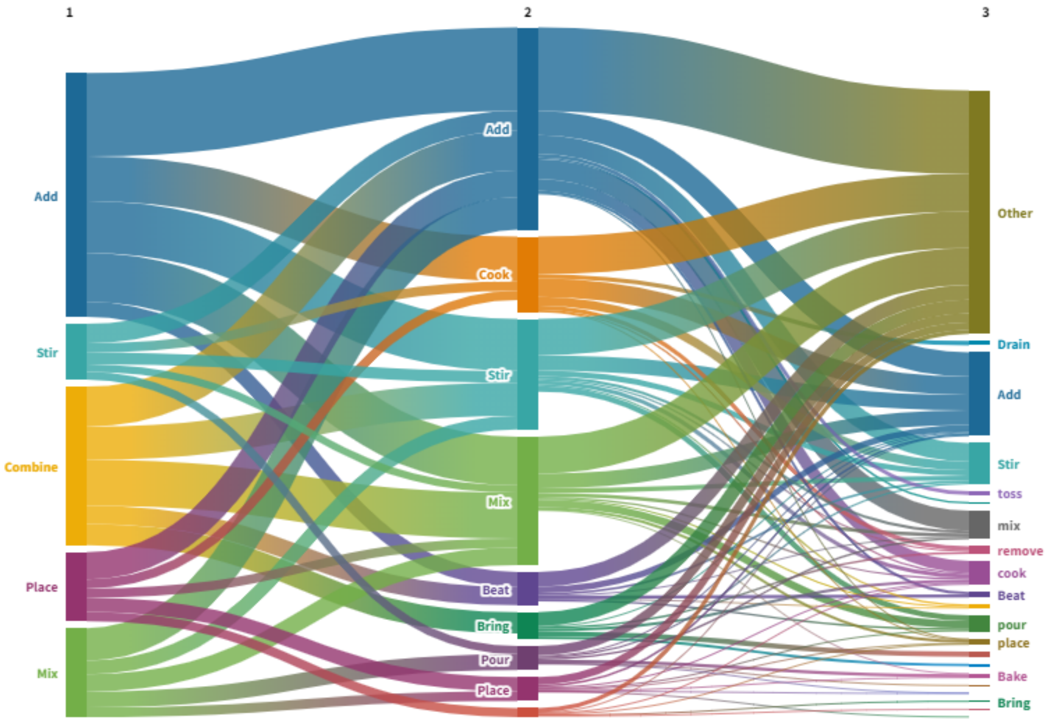

FIGURE 5.11: Five Frequently Used Starting Recipe Actions along with Frequently used following Actions.

Transformation of sentences to actions and ingredients was a challenging task (as discussed in section 5.1.4) and resulted in 10638183 actions and 8855411 ingredients. Figure 5.11 shows a frequently used combination of actions in 0.8M-Recipes. This shows frequent actions used among recipes. Based on the position of ingredients and actions in a sentence, we established a pairing between action and ingredients. Let pair $(a_i, ings_i)$ specifies that action a_i will be performed on ingredients $ings_i$ then $position(a_i) < position(ings_i)$ and $\forall_j \{position(ings_i) - position(a_i) < position(ings_i) - position(a_j)\}$ where $j = 1 \cdots k$ and $i \neq j$. There would be a M-M (i.e. many to many) mapping relationship between ingredients and actions. Figure 5.12 shows a correlation between frequent ingredients and frequent actions. This also gives an idea of permissible actions on ingredients (e.g. spread action can be performed on butter but beating, sauteing, or pouring is not permissible on butter).

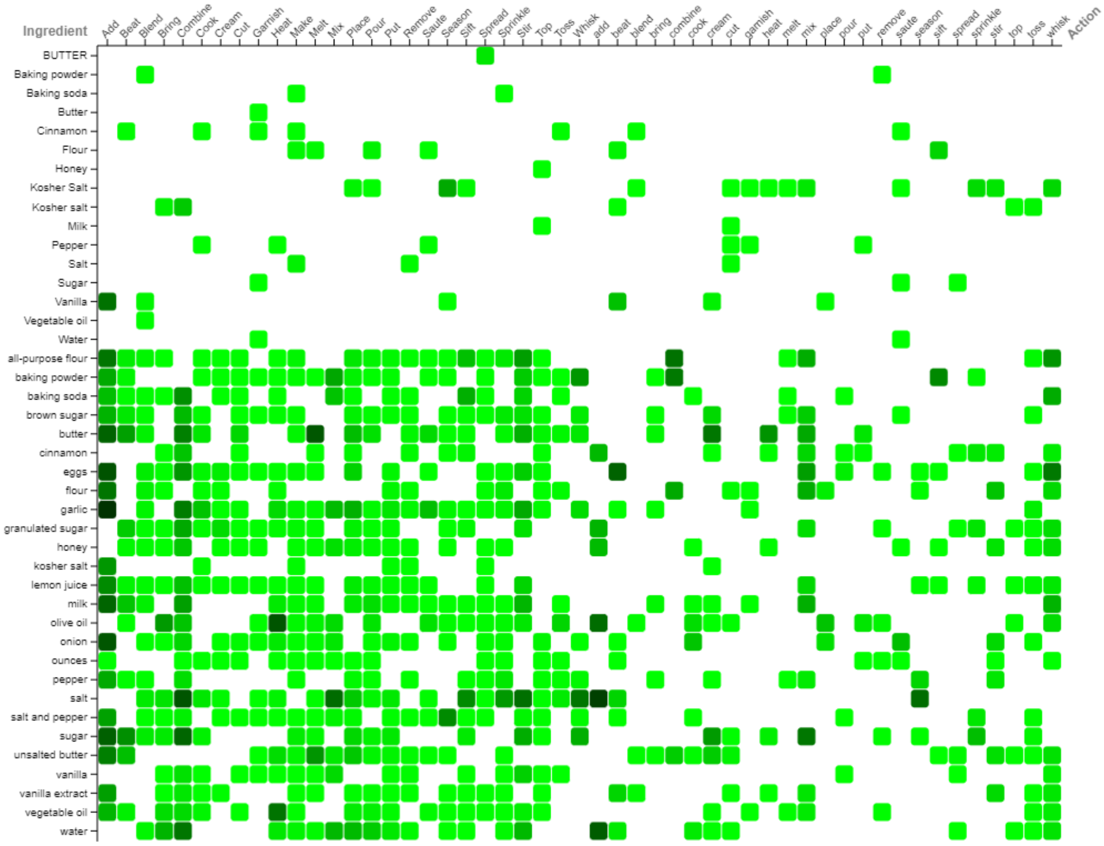

FIGURE 5.12: Correlation Matrix Between 25 Frequently used Ingredients and 25 Frequently used Starting Actions. Higher Color Intensity Represents strong Co-Relation

There were a few issues faced while populating *Instructions* and mapping sentences to actions and ingredients. These include

Lexical variation of instruction sentences Instruction sentences comprise a mix of simple, complex, and compound sentences. A simple sentence consists of an action and an ingredient (e.g. Cook the rice noodles according to the packet instructions). A complex sentence consists of an action and more than one ingredient (e.g. Beat the feta and capers together to form a rough paste). While a compound sentence consists of multiple actions and a subset of recipe ingredients with each action. These actions are connected through conjunction operators (e.g. Transfer the carrots to a roasting tin and drizzle with maple syrup or honey and avocado oil). This lexical variation in instruction sentences resulted in an increased number of regular expressions to accurately extract actions and ingredients.

Missing ingredients 0.8M-Recipes also has instruction sentences that contain action but have missing ingredients. In such cases, ingredients were imputed from the last in-sequence sentence. Consider the two sentences given in the following list. The former sentence has an action "toss" with the ingredient "Worcestershire sauce". While the later sentence has only the action "stir" without any ingredient. Hence, we impute the ingredient "Worcestershire sauce" in the later sentence as well.

- Toss in the Worcestershire sauce, then turn the heat off.
- Stir well, season to perfection, and serve.

5.1.5.4 Entity: Nutrition

Food is a major source of nutrition for the human body. A balanced and healthy diet is necessary to maintain a balance of nutrients in the body. Excess of any nutrient can create certain health conditions (like excess of sodium can lead to high blood pressure, heart attack, or stroke). Similarly, deficiency of any nutrient can also affect body functions (deficiency of potassium results in hypokalaemia that limits nerve and muscle functions).

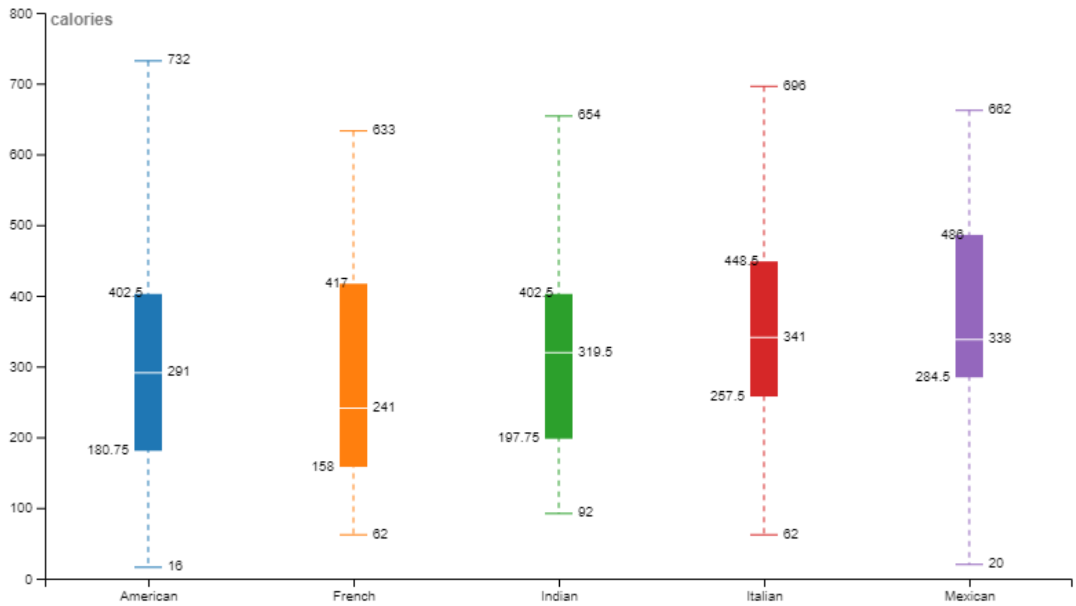

FIGURE 5.13: Average Level of Calories in Different Cuisines.

Figure 5.13 shows the level of calories among different cuisines. Here we can see Mexican and Italian recipes have the highest calories level while French recipes have the lowest calories level.

5.2 RecipeKG: A Semantically-Driven Knowledge Graph to Manifest Culinary Recipes

The late 20th century was a revolutionary period of culinary food that shifted cooking recipes from cooking books and TV shows to internet-based recipe websites, blogs, and forums. Publishing an online recipe is much easier than sharing it on TV shows or cooking books and hence resulted in a steady increase in online recipes in the last few decades. Online recipes though semi-structured but supports machine readability. Immense recipe volume and machine readability of online recipes create an opportunity to organize, link, and explore recipe related knowledge to create machine-understandable recipes. These machine-understandable recipes would be helpful in finding alternative recipes (i.e. using alternative ingredients or alternative actions) and generating novel recipes (i.e. using alternative procedures). This novelty in recipe generation would create new possibilities (for food enthusiasts, sick persons, or food-conscious people) to evolve recipes using the knowledge extracted from existing recipes.

A knowledge graph is a graph-structured knowledge base that interlinks the entities through well-defined semantic descriptions. Several food knowledge graphs have been proposed with specialized scope and application areas. Flavorgraph (Park et al., 2021) was created to find the food pairing based on the flavor molecule information in recipes. FoodKG (Haussmann et al., 2019) was created to serve as an aggregated knowledge base to recommend healthier food items based on the user's health condition and taste preferences. Lei, Zhenfeng, et al created RcpKG (Lei et al., 2021) based on multi-modality and hierarchical data to generate novel recipes. A knowledge graph based on concepts and relationships related to food production and food inspection was covered by food safety knowledge graph (Qin et al., 2019). Food knowledge graph (Rostami et al.,

2021) proposed by Qin, Li, Zhigang Hao, and LiPing Yang establishes relationships between food and ingredient based on data obtained from recipes and nutrition databases. Food spot-check knowledge graph (Qin et al., 2020) and cross-modal knowledge graph (Chen et al., 2017) were created to answer user questions related to food safety and food additives respectively. Trophic knowledge graph (Le Guillarme et al., 2021) is created using a pipeline that extracts, normalizes, and aggregates diet and nutritional information. Diet recommendations based on personal health conditions using a knowledge graph have also been proposed by Seneviratne, Oshani, et al. (Seneviratne et al., 2021). In recent attempts, Min, Weiqing, et al. explores the potential of knowledge graph in food intelligence and the internet of food (Min et al., 2022), while Wang, Di, et al. creates a cross-modal knowledge graph to predict ingredients and actions of the culinary recipe using food image as input (Wang et al., 2022).

Ontology is the backbone of a knowledge graph that defines the concepts and relationships used to express the data in the knowledge graph. Food knowledge graphs are also based on food ontologies (e.g. FoodKG (Haussmann et al., 2019) is based on WhatToMake [14] ontology (Qi et al., 2018), Food spot-check knowledge graph (Qin et al., 2020) and food safety knowledge graph are both based on food safety ontology (Qin et al., 2019), Food knowledge graph (Rostami et al., 2021) is based on foodon [15] ontology (Dooley et al., 2018)). Several food ontologies exist that power various knowledge graphs but all of them (except *RecipeOn* [16]) lack in their semantic representation of a recipe as a process and deals the recipes as a knowledge base of ingredients and instructions. On the other hand, *RecipeOn* treats recipes both as a knowledge base and a process. As a process, *RecipeOn* keeps track of the actions applied on an ingredient and sequence of actions using $evo : hasAction$ and $seq : directlyFollows$ relationships respectively.

RecipeOn ontology is divided into five main components (i.e. CoreRecipe, Ingredients, Actions, Procedures, and Nutrition) with a pre-defined scope based on the functional requirements. It has 231 classes/concepts with hierarchical

[14] https://raw.githubusercontent.com/foodkg/foodkg.github.io/master/ontologies/WhatToMake.rdf

[15] https://www.ebi.ac.uk/ols/ontologies/foodon

[16] https://github.com/HajiraJabeen/EvoRecipesOntology/blob/main/EvoRecipes.owl

division. The ingredient class hierarchy of *RecipeOn* has taken its inspiration from the ingredient taxonomy of "The Cook's Thesaurus" [17]. The concepts of *RecipeOn* are linked using 37 object properties. Moreover, *RecipeOn* also includes 20 data properties.

In this chapter, we have proposed a Recipe Knowledge Graph (*RecipeKG*) that is based on the semantic concepts and relationships from *RecipeOn* ontology. *RecipeKG* is semantically rich in recipe representation compared to its predecessors in the food domain. It comprises of information about recipe features, plurality of recipe metadata, provenance, nutrition, ingredients, actions, sequence of actions, and procedures comprising ingredients and actions. *RecipeKG* is a large-scale semantic-aware recipe knowledge graph that is created to increase the machine-understandably of the recipe process. It contains 0.8 million recipe nodes, 8.9 million ingredient nodes, 10.8 million action nodes, 2.1 million nutrient nodes, and 209 million facts. A summary of *RecipeKG* entities is shown in table 5.4, while table 5.5 shows the number of facts against various relationships.

TABLE 5.4: Descriptive Statistics of *RecipeKG* Entities

Module	Concept	Instances
CoreRecipe	schema:Recipe	810800
	schema:Person	96489
	schema:AggregateRating	351383
	qudt:Quantity	786278
	qudt:QuantityValue	786278
	qudt:Unit	786278
Ingredients	evo:Ingredient	8855411
	evo:IngredientType	8855411
	qudt:Quantity	7746831
	qudt:QuantityValue	7279846
	qudt:Unit	6128330
Actions	seq:Action	10638183
Procedures	evo:Procedure	142672
Nutrition	schemaNutritionInformation	234584
	evo:Nutrient	2139035
	qudt:Quantity	2139035
	qudt:QuantityValue	2139035
	qudt:Unit	2139035

RecipeKG pipeline comprises three stages (i.e. data retrieval, canonicalization, and knowledge graph generation) as shown in figure 5.14 that transforms the web-based semi-structured recipe data to machine-understandable facts (i.e.

[17] http://www.foodsubs.com

<entity, relationship, entity>). Finally, an end user can query *RecipeKG* using SPARQL to retrieve a subset of facts.

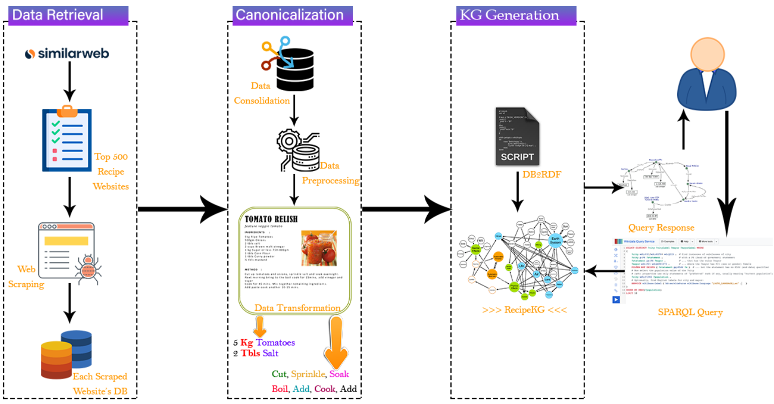

FIGURE 5.14: *RecipeKG* Pipeline Comprising of Three Stages

The data Retrieval stage comprises of three activities (i.e. selection of top recipe websites, web scraping, and storage of each website's scrapped data to a separate database) as discussed in section 5.1.1. In the data canonicalization stage, we first consolidate the data (section 5.1.2) to a single database, then we pre-process (section 5.1.3) the data, and finally transform the data (section 5.1.4) to a format that is suitable for knowledge graph creation. In the final stage (i.e. KG Generation) we have written a script that maps the data from the relational database to the entities and relationships of the knowledge graph. Figure 5.15 shows the level 1 relationship of a recipe with other nodes in *RecipeKG*, figure 5.16 shows the level 2 relationship of a recipe with ingredients using hasIngredient relationships in *RecipeKG* and figure 5.17 shows the level 3 relationship of action nodes with hasAction and directlyPrecedes relationships in *RecipeKG*. These levels indicates how detailed relationships are described in *RecipeKG* knowledge graph. There are five modules in *RecipeOn* ontology (core recipe, ingredient, action, procedure, and nutrition). Each module comprises of various relationships and a large number of facts have been generated against each relationship in *RecipeKG* like core recipe has 0.3 million facts against evo:hasAggregateRating Tables 5.5, 5.6, 5.7, 5.8, and 5.9 list the number of facts against each module in *RecipeKG*

FIGURE 5.15: Level-1 Nodes and Relationships of a Recipe in *RecipeKG*. The Recipe is the Root Node while all other Nodes are Level 1 Nodes with Corresponding Relationships from the Recipe Node.

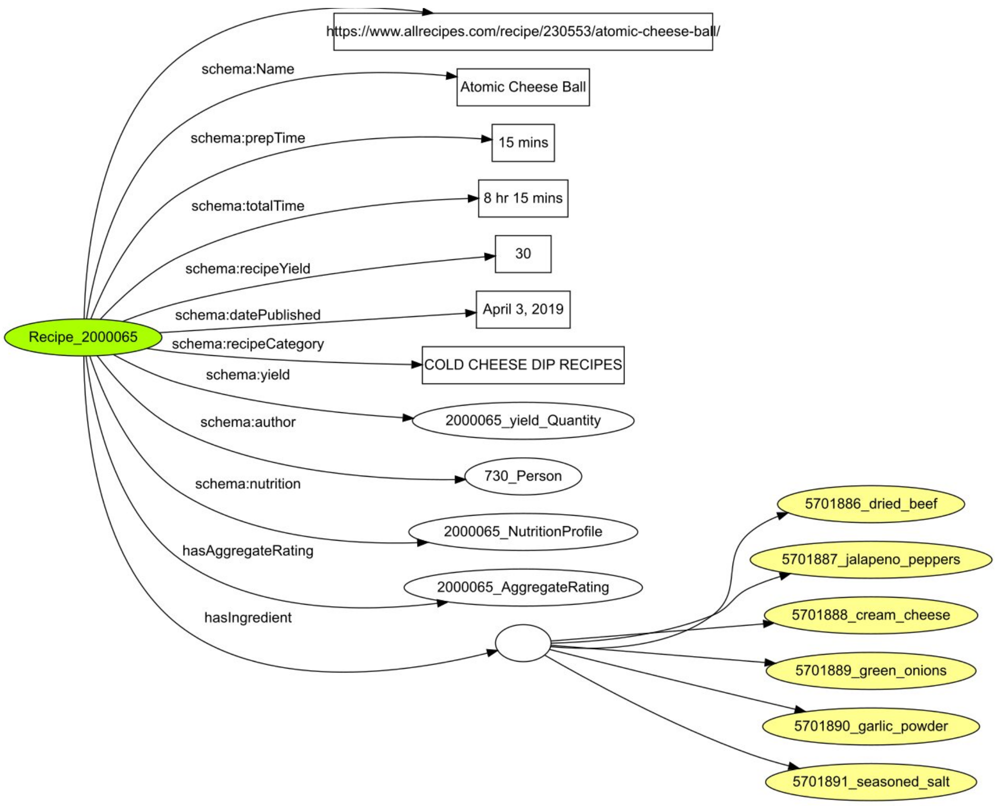

FIGURE 5.16: Level-2 Nodes and Relationships of a Recipe in *RecipeKG*. Level-2 Ingredient Nodes in Yellow Color are Linked from the Recipe Node using a Blank Node with hasIngredient Relationship.

Table 5.5: Number of Facts in *RecipeKG* against various Semantic Relationships (as defined by *RecipeOn*) of Core Recipe Module.

Relationship	Domain	Range	Facts
rdf:type	Recipe Instance	schema:Recipe	810800
evo:hasAggregateRating	schema:Recipe	schema:AggregateRating	351383
schema:author	schema:Recipe	schema:Person	622190
schema:prepTime	schema:Recipe	qudt:Quantity	306800
rdf:type	Quantity Instance	qudt:Quantity	786278
schema:cookTime	schema:Recipe	qudt:Quantity	273836
rdf:type	QuantityValue Instance	qudt:QuantityValue	786278
schema:totalTime	schema:Recipe	qudt:Quantity	333635
rdf:type	Unit Instance	qudt:Unit	786278
schema:yield	schema:Recipe	qudt:Quantity	252721
schema:nutrition	schema:Recipe	schema:NutritionInfo.	234584
rdf:type	AggregateRating Instance	schema:AggregateRating	351383
evo:hasIngredient	schema:Recipe	evo:Ingredient	8951688
evo:hasProcedure	schema:Recipe	evo:Procedure	142672
rdf:type	Person Instance	schema:Person	96489
qudt:hasQuantityValue	qudt:Quantity	qudt:QuantityValue	786278
qudt:hasUnit	qudt:Quantity	qudt:Unit	786278
schema:name	schema:Recipe	xsd:string	810731
evo:hasCourseName	schema:Recipe	xsd:string	38268
evo:hasDifficultyLevel	schema:Recipe	xsd:string	139353
datePublished	Recipe	xsd:string	278918
keywords	schema:Recipe	xsd:string	327149
recipeCategory	Recipe	xsd:string	241956
recipeYield	Recipe	xsd:integer	545195
url	Recipe	xsd:string	810800
givenName	Person	xsd:string	96489
ratingCount	Recipe	xsd:integer	251008
ratingValue	Recipe	xsd:integer	208543
reviewCount	Recipe	xsd:integer	278051
qudt:hasNumericValue	qudt:QuantityValue	xsd:integer	786278
qudt:abbreviation	qudt:Unit	xsd:string	786278

Table 5.6: Number of Facts in *RecipeKG* against various Semantic Relationships (as defined by *RecipeOn*) of Ingredient Module.

Relationship	Domain	Range	Facts
rdf:type	Ingredient Instance	owl:Ingredient	8855411
evo:hasType	evo:Ingredient	evo:IngredientType	8855411
rdf:type	Quantity Instance	qudt:Quantity	7746831
evo:hasAction	evo:Ingredient	owl:Action	5390211
qudt:hasQuantityValue	qudt:Quantity	qudt:QuantityValue	7468799
qudt:hasUnit	qudt:Quantity	qudt:Unit	6128330
rdf:type	QuantityValue Instance	qudt:QuantityValue	7279846
rdf:type	Unit Instance	qudt:Unit	6128330
schema:name	schema:Ingredient	xsd:string	8855411
qudt:hasNumericValue	qudt:QuantityValue	xsd:integer	7279846
qudt:abbreviation	qudt:Unit	xsd:string	6128330

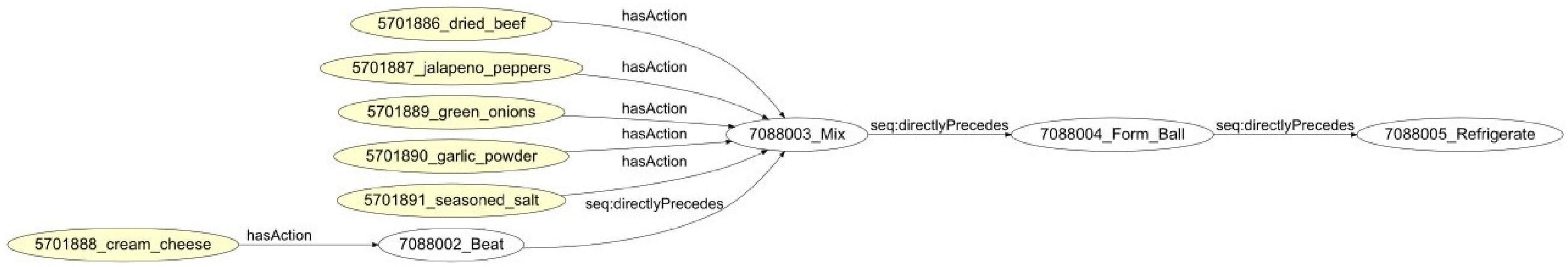

FIGURE 5.17: Level-3+ Action Nodes with hasAction and directlyPrecedes Relationships.

Table 5.7: Number of Facts in *RecipeKG* against arious Semantic Relationships (as defined by *RecipeOn*) of action module.

Relationship	Domain	Range	Facts
rdf:type	Action Instance	owl:Action	10638183
seq:directlyPrecedes	owl:Action	owl:Action	9328423
schema:name	owl:Action	xsd:string	10638183

Table 5.8: Number of Facts in *RecipeKG* against various Semantic Relationships (as defined by *RecipeOn*) of Procedure Module.

Relationship	Domain	Range	Facts
rdf:type	Procedure Instance	evo:Procedure	142672
evo:hasProcedureIngredient	evo:Procedure	evo:Ingredient	910134
evo:hasProcedureAction	evo:Procedure	owl:Action	222435
schema:name	owl:Action	xsd:string	142672

Table 5.9: Number of facts in *RecipeKG* against various semantic relationships (as defined by *RecipeOn*) of nutrition module.

Relationship	Domain	Range	Facts
rdf:type	NutritionInfo. Instance	owl:NamedIndividual	234584
rdf:type	NutritionInfo. Instance	evo:Procedure	234584
evo:hasNutrientType	schema:NutritionInfo.	evo:Nutrient	2139035
evo:hasNutrientQuantity	evo:Nutrient	qudt:Quantity	2139035
rdf:type	Quantity Instance	owl:NamedIndividual	2139035
rdf:type	Quantity Instance	qudt:Quantity	2139035
qudt:hasQuantityValue	qudt:Quantity	qudt:QuantityValue	2139035
qudt:hasUnit	qudt:Quantity	qudt:Unit	2139035
rdf:type	QuantityValue Instance	owl:NamedIndividual	2139035
rdf:type	QuantityValue Instance	qudt:QuantityValue	2139035
rdf:type	Unit Instance	owl:NamedIndividual	2139035
rdf:type	Unit Instance	qudt:Unit	2139035
schema:name	owl:Action	xsd:string	2139035
qudt:hasNumericValue	qudt:QuantityValue	xsd:integer	2139035
qudt:abbreviation	qudt:Unit	xsd:string	2139035

Finally, to manipulate and query the RDF data, we loaded *RecipeKG* in Blazegraph [18] (SYSTAP, 2019). Blazegraph is a graph database tool that supports RDF data and provides the facility to explore knowledge graphs using SPARQL queries. Blazegraph provides scalability and can store upto 50 billion facts.

[18] https://blazegraph.com/

5.3 Use cases and Discussion

5.3.1 Recipe Profile

Each recipe has a profile that comprises a rich set of relationships to provide information about the recipe process, metadata, and provenance. This helps users to find the desired recipes ranging from simple search queries (like show me dessert recipes) to advanced search queries(like What are the gluten-free recipes that can be prepared in 15 minutes?). A few competency questions that relate to the recipe profile are as follows.

- Which recipes have high calorie content (more than 300)?
- List down the vegetarian recipes.
- Which recipes contain baked potatoes

Listing 5.1 shows the SPARQL Query relevant to the first competency question.

LISTING 5.1: SPARQL Query to List Recipes having High Calories (more than 300)

```
PREFIX : <https://hajirajabeen.github.io/EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe <http://schema.org/nutrition> ?nutrinfo.
?nutrinfo :hasNutrientType :CalorieContent.
:CalorieContent a :Nutrient.
:CalorieContent :hasNutrientQuantity ?quan .
?quan qudt:hasQuantityValue ?qv .
?qv qudt:hasNumericValue ?nv
FILTER (?nv > "300"^^xsd:double)
}
```

5.3.2 Recipes for Sick and Diet Conscious People

Healthy food is important for sick and health-conscious people as it improves both physical and mental health. A food that is rich in nutrients not only helps to improve mood and avoids depression but also improves immunity for fast recovery and avoids chronic diseases. Consuming a balanced diet helps manage chronic diseases like chronic kidney disease, hypertension, and diabetes. Moreover, a suitable diet provides the body necessary nutrients that can support quick recovery from the disease. Also, it helps to improve the immune system.

For diet-conscious persons, a nutrient-dense diet helps to improve overall well-being. This helps individuals to perform daily activities easily. A balanced diet also supports maintaining a healthy weight and also helps to avoid chronic diseases.

A few competency questions related to sick and diet-conscious people are as follows.

- Which rice recipes are suitable for patients having gastric ulcers (less spicy foods)?
- Which recipes are suitable for heart patients (low cholesterol recipe)?
- Which recipes contain boiled chicken?

Listing 5.2 shows the SPARQL query against the first competency question mentioned above.

LISTING 5.2: SPARQL Query to find what are the Recipes suitable for Patients having Stomach Ulcers

```
PREFIX : <https://hajirajabeen.github.io/EvoRecipesOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/2.1/schema/qudt#>

SELECT DISTINCT ?recipe
WHERE { ?recipe a <http://schema.org/Recipe> .
?recipe :hasIngredient :Rice .
?recipe :hasIngredient ?ingredient .
FILTER NOT EXISTS{
?ingredient :hasFlavoringIngredient ?flavor .}
```

```
}
```

5.4 Summary

In this chapter, we have proposed a 0.8M-Recipes dataset that comprises of 50+ attributes. The 0.8M-Recipe dataset has been curated after data extraction, data consolidation, data preprocessing, and data transformation phases. On top of 0.8M-Recipe, we have generated a large-scale semantic-aware Recipe Knowledge Graph *RecipeKG* that increases the machine-understandably of the recipe process. *RecipeKG* contains 0.8 million recipe nodes, 8.9 million ingredient nodes, 10.8 million action nodes, 2.1 million nutrient nodes, and 209 million facts. Process representation helps to create machine-generated recipes using appropriate alternative actions and ingredients. *RecipeKG* provides ingredient-ingredient, ingredient-action, and action-action relationships using the concepts and properties from *RecipeOn* ontology. Moreover, we have also presented usecases of recipe profiling and recipes for sick & diet-conscious people.

Chapter 6

EvoRecipes: A Generative Approach for Evolving Context-Aware Recipes

Culinary recipe creation is both an art and a science. The artistic aspect of a recipe involves the texture, taste, aroma, and presentation that require imagination, technical skills, and creativity. Just like artists, chefs express their understanding of traditional recipes, cuisines, and unique cooking styles by combining and replacing ingredients, assessing flavors, balancing ingredient quantities, and appealingly presenting food. On the other hand, the scientific aspect requires knowledge of physics, biology, and chemistry. The scientific aspect also involves the chemical reactions of ingredients (e.g., Maillard reaction, responsible for the browning of food and the development of flavor), the impact of biological microorganisms on food (e.g., yeast in bread baking, bacteria in fermentation processes, and enzymes in the ripening of fruits and vegetables), and the impact of physical processes (e.g., roasting, steaming, and boiling) on the flavor and texture of food. Recipe creation is based on experience, imagination, as well as an understanding of scientific principles and processes, which proves that cooking is both an art and a science.

Creativity in culinary recipes has gained more importance in recent years due to the increasing interest of people in food culture and memorable dining experiences. Demand for specialized recipes from sick persons, diet-conscious people, and food enthusiasts also complements culinary creativity. Moreover, the rise of the social web has made it easier for chefs to share their newly generated recipes (based on their experiences, intuition, & imagination) and get recognition for their culinary creativity. However, human creativity in culinary recipes has a few limitations: *(i)* Error - Humans may produce errors while creating a new recipe or may misjudge the outcome of a recipe; *(ii)* Biasness - Chefs may have biases for certain ingredients based on their availability in a certain region or certain recipes that are well received by the customers. Similarly, chefs may have preferences for certain cooking methods based on their skills limitations, or experiences. These factors play a significant role in limiting creativity; *(iii)* Speed - Human chefs need more time and effort to generate novel recipes than computational alternatives. On the other hand, computational creativity in culinary recipes overcome human errors, is not affected by biases, and can generate recipes at a much faster pace than humans.

Food is considered a computational artifact (Deng et al., 2022) and researchers are exploring the potential of computational creativity to generate novel recipes (Antô et al., 2020, Jabeen et al., 2020, Loughran and O'Neill, 2017). Also, with the evolution of the semantic web, machine-understandable recipes have gained more attention in recent years. Several food ontologies (e.g. FoodOn[1] (Dooley et al., 2018), RecipeOn[5]) and knowledge graphs (e.g. FlavorGraph[2] (Park et al., 2021), FoodKG[3] (Haussmann et al., 2019), RecipeKG[4]) have been developed in the last decade that facilitates interoperability and increases machine-understandability for the food domain. Creativity in recipes involves multiple techniques that include combining an unusual group of ingredients, replacing ingredients, or attempting alternative cooking methodologies. These techniques are implemented using various models including statistical language models (Antô et al., 2020) and transformers (Gim et al., 2021).

The main objective of this chapter is to explore computational creativity in culinary recipes using evolutionary algorithms. Therefore, we have proposed an evolutionary framework $EvoRecipes$ that uses Genetic Algorithm (GA) (Chambers, 2019, Mitchell, 1998) to generate novel recipes (in RDF format) based on user preferences. Quantitative and Qualitative evaluation functions are designed to check the alignment (of recipes with user preferences) and quality of novel recipes that have been generated using the $EvoRecipes$ framework. $EvoRecipes$ generates novel recipes by exploring different recipe components (such as ingredients, actions, and procedures). An initial population of context-aware recipe solutions is initialized from the $RecipeKG$ knowledge graph. $RecipeKG$

[1] https://foodon.org/
[2] https://github.com/lamypark/FlavorGraph
[3] https://foodkg.github.io/
[4] https://hajirajabeen.github.io/RecipeKG

encodes human-generated recipes in RDF format and is based on a 0.8 M recipes dataset. *EvoRecipes* is based on the structure of recipes proposed in *RecipeOn* ontology and interacts with the machine-understandable recipes. Finally, to improve the human understandability of novel recipes, we have used OpenAI GPT API (Davinci model) to generate recipe text from RDF format for newly generated recipes. In this article, we have the following contributions:

- *EvoRecipes* framework to evolve culinary recipes.
- *EvoRecipes* encodes recipes in RDF format (using classes and properties as defined in the *RecipeOn* ontology).
- *EvoRecipes* uses GA to explore the large solution space of encoded recipe solutions and is capable of incorporating user preferences.
- *G*A is used to create novel recipes using the following recipe evolutionary operators
  - Mutation: Ingredient Replacement
  - Mutation: Action Replacement
  - Mutation: Action Interchange
  - Crossover: Alternative Procedure
- Proposed multi-objective fitness function to evaluate the quality of novel recipes.
- Proposed qualitative metrics to evaluate the subjective parameters of novel recipes.
- Recipe RDF to Recipe Text Generation using OpenAI GPT.

This chapter will cover the following research questions.

RQ4. How can computational techniques be used to generate context-aware novel recipes that are also capable to accommodate user preferences?

RQ5. How can we evaluate the quality of machine-generated novel recipes?

This chapter is based on the following article.

Muhammad Saad Razzaq, *Fahad Maqbool, Muhammad Ilyas, Hajira Jabeen, A Generative Approach for Evolving Context-Aware Recipes, published in IEEE Access, vol. 11, pp. 74148-74164, 2023, doi: 10.1109/ACCESS.2023.3296144.*

The chapter is organized as follows. In section 6.1 we have briefly explained the *RecipeOn* ontology and *RecipeKG* knowledge graph. In section 6.2, *EvoRecipes* framework and its components have been discussed in detail. Qualitative recipe evaluation metrics are discussed in 6.3, we have discussed qualitative evaluation metrics. Experimental setup, experiment detail, and comparison with related techniques are discussed in section 6.4. Finally, we present the summary in section 6.5.

6.1 Preliminaries

6.1.1 RecipeOn

RecipeOn[5] is a recipe ontology that not only helps to increase machine understandability but also encodes important information about the recipe. It includes ingredients, actions, nutrition, and sequencing of instructions. *RecipeOn* ontology guides the user in preparing recipes as a systematic process. It facilitates not only providing information related to ingredients and actions but also the sequence of actions and ingredients related to each procedure. *RecipeOn* helps evolve new recipes and personalize the existing recipes for different user preferences using alternative ingredients and alternative action relationships. Furthermore, *RecipeOn* can map recipes as a process.

6.1.2 RecipeKG

A Recipe Knowledge Graph *RecipeKG* [16] is built using 0.8M-Recipes[15] that comprises 0.8 million human-generated recipes and collected from top-ranked recipe websites. It contains basic recipe details, a list of ingredients, nutritional information, and instruction statements. *RecipeKG* uses the concepts

[5] https://www.allrecipes.com/recipe/8453018/easy-air-fryer-whole-chicken/

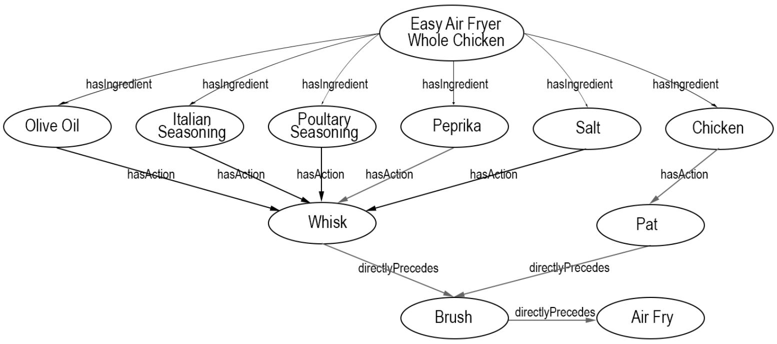

FIGURE 6.1: "Easy air fryer whole chicken"[5]recipe represented as a process based on classes and relationship of RecipeOn ontology.

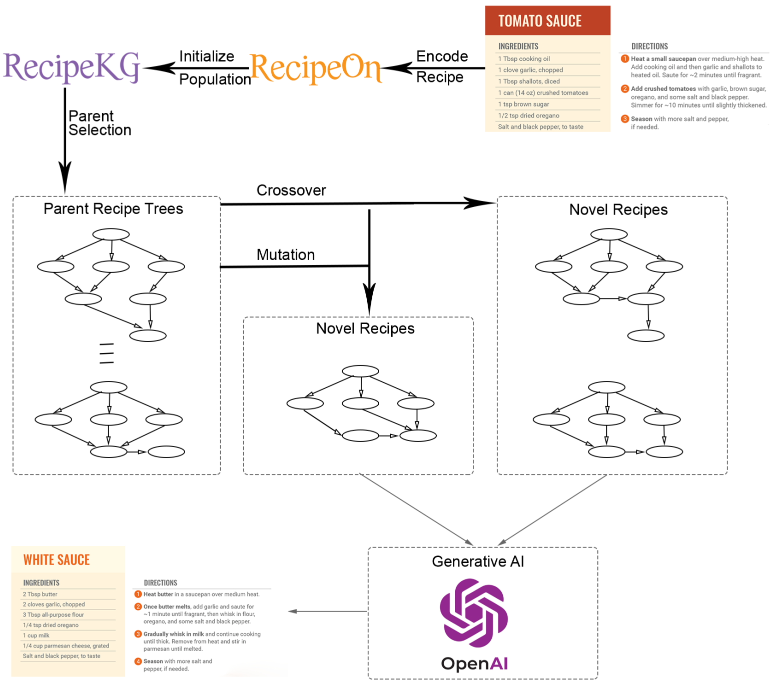

FIGURE 6.2: EvoRecipes framework generates new semantic-aware recipes (encoded in RDF format) using evolutionary operators and generates human-readable recipe text using the Davinci model of OpenAI GPT API

and relationships as specified in *RecipeOn* ontology. It represents the recipe as a process and defines the ingredient-action, action-action, and ingredient-action relationships. *RecipeKG* is based on a huge collection of 52.96 million instances and 209 million facts.

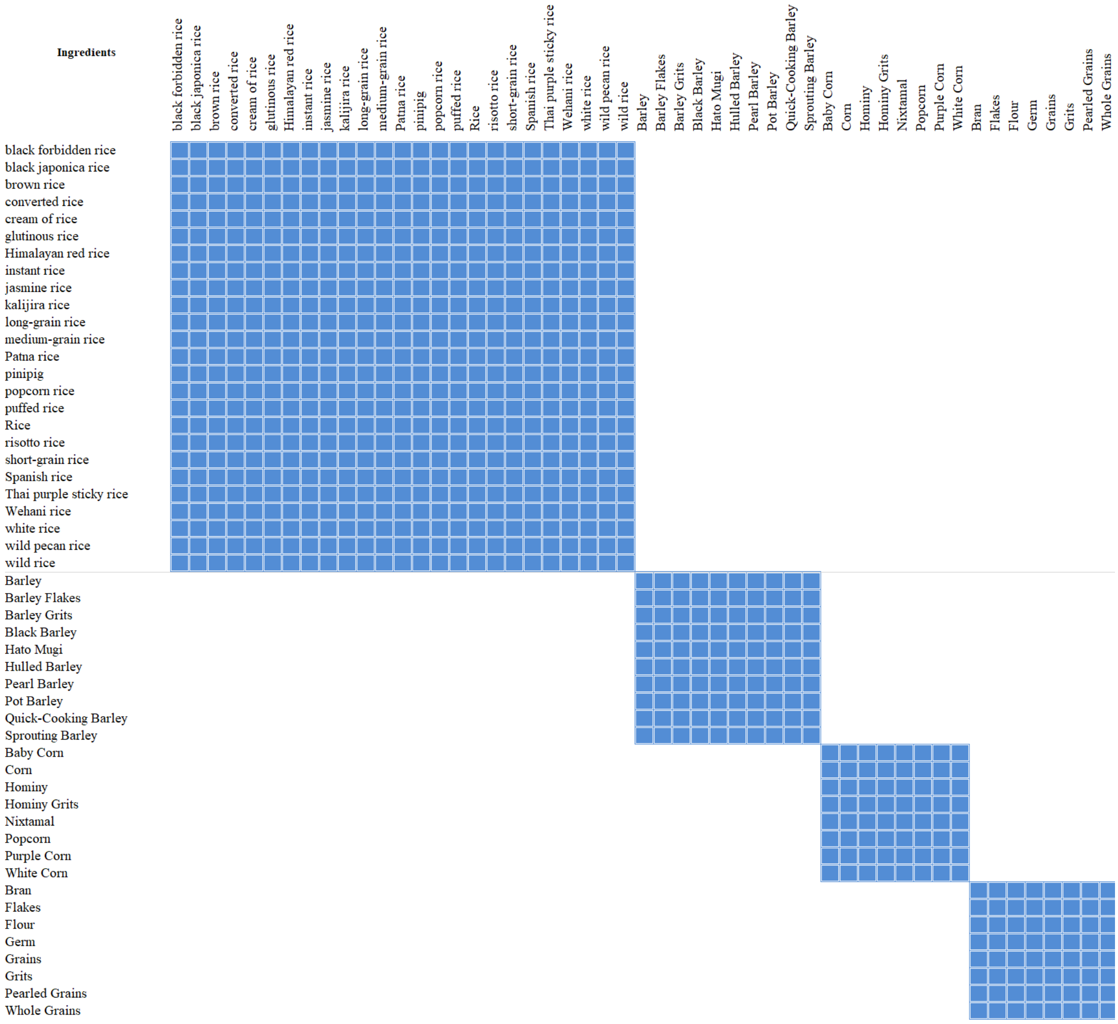

FIGURE 6.3: Alternative Ingredient Against Four Ingredient Classes (i.e. Rice, Barley, Corn, and Grain Types)

6.2 EvoRecipes: A Generative Approach for Evolving Context-Aware Recipes

EvoRecipes is a recipe evolution framework that generates customized recipes based on user preferences. It has been built making use of *RecipeOn* ontology that represents the recipe as a process as shown in figure 6.1. Ingredients are categorized into three ingredient types (i.e. $MainIngredient$, $SideIngredient$, and $FlavorIngredient$) using three different object properties (i.e. $hasMainIngredient$, $hasSideIngredient$, and $hasFlavoringIngredient$). Each ingredient has an object property $hasAction$ that determines the action performed on each ingredient.

Actions have some sequence among themselves that is ensured by $RecipeOn$'s object properties (i.e. $seq : directlyFollows$ and $seq : directlyPrecedes$).

In this article, we have proposed a $EvoRecipes$ framework (figure 6.2) that uses both evolutionary algorithm and generative AI to create custom context-aware and human-readable recipes. Evolutionary algorithms are based on the phenomena of survival of the fittest and calculate the fitness of an individual using a fitness function. Therefore, fitness functions are tuned to accommodate user choices and preferences. Moreover, recipes have been encoded using $RecipeOn$ ontology which makes them not only machine-readable but also machine-understandable.

$EvoRecipes$ starts with a population of initial solutions (RDF encoded recipes) retrieved from $RecipeKG$ [16]. It then selects the parent solutions and applies evolutionary operators (i.e. mutation, crossover) to generate offspring. The process of parent selection and evolution repeats for a certain number of iterations until the stopping criteria are met. The resultant solutions (RDF encoded recipes) are finally transformed into recipe text using the Davinci model of OpenAI GPT API. $EvoRecipes$ comprises several components (such as initial population, selection, mutation, crossover, and fitness function evaluation) that have been discussed in the following subsections.

6.2.1 Initial Population

We select the initial population of recipes from $RecipeKG$ that has a collection of recipes stored in RDF format. $RecipeKG$ is based on a 0.8M-Recipes dataset. The dataset comprises 0.8 million real human-generated recipes extracted from recipe websites.

6.2.2 Selection

We have used the roulette wheel as a population selection criterion in $EvoRecipes$. This is the fitness proportionate population selection criteria for reproduction as it gives a chance to individuals with low fitness value for being selected in the next generation along with more fit individuals. Hence it ensures population diversity by selecting individuals with varying fitness values.

6.2.3 Mutation

Mutation replaces existing ingredients and actions in recipes with choices that not only add diversity to the population but also improve the novelty. In *EvoRecipes* we have proposed three mutation techniques that modify both ingredient and action nodes. These techniques have been discussed in detail in the following subsections.

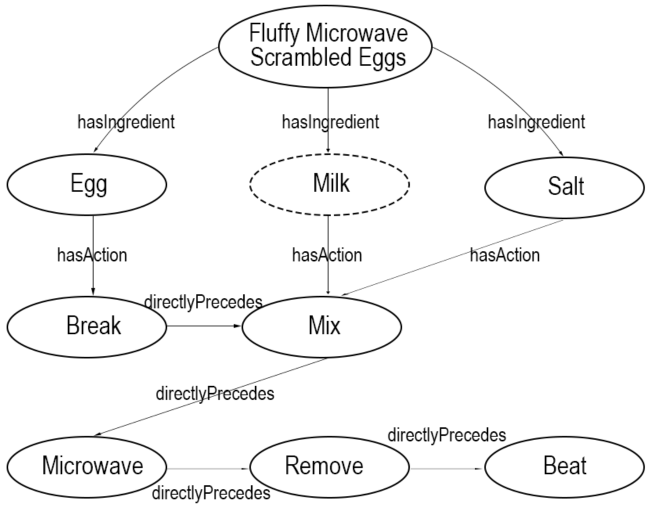

FIGURE 6.4: Fluffy Microwave Scrambled Eggs Available at AllRecipes

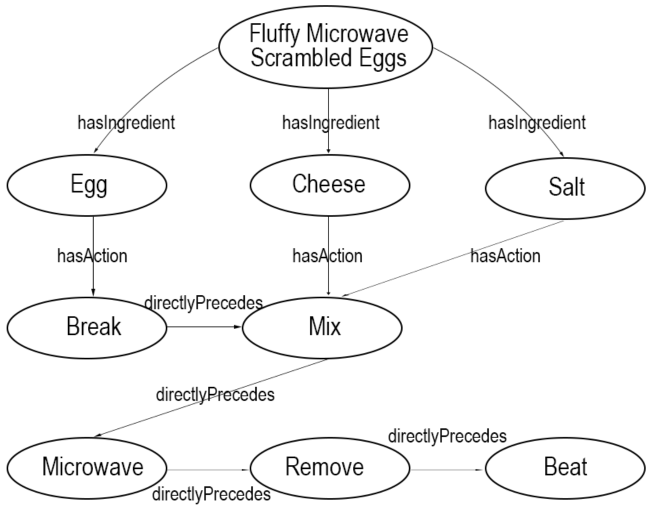

FIGURE 6.5: Mutated (Milk Substitution with Cheese) Fluffy Microwave Scrambled Eggs

6.2.3.1 Ingredient Substitution

The ingredient is one of the key components of a recipe. It has a vital role in recipe preparation, taste, color, aroma, and presentation. Some ingredients have a primary role and are classified as $MainIngredients$, while others have a secondary role and are classified as $SideIngredient$ or $FlavoringIngredient$. $RecipeOn$ maintains an ingredient class hierarchy and divides ingredients into 12 classes and numerous sub-classes. Ingredient substitution replaces an ingredient i belonging to class c with another ingredient j that also belongs to the same class c. For example, nixtamal can be replaced with popcorn as both belong to the same corn class. Figure 6.3 shows the ingredients and their alternatives from four classes(i.e. rice, barley, corn, and grain types). Due to space limitations, we have not mentioned the remaining ingredient classes, however, their complete detail is available in $RecipeOn$ ontology.

Figure 6.4 and figure 6.5 shows an ingredient substitution in which "fluffy microwave scrambled Eggs"[6] (figure 6.4) is mutated to generate a new recipe (figure 6.5) by replacing milk with cheese.

6.2.3.2 Action Substitution

The same ingredients but different actions can lead to very different recipes. It is impossible to prepare a food item without appropriate action details. $RecipeOn$ divides $Action$ class into three sub-classes (i.e. $Cooking$, $Preparatory$, and $PostCooking$). In the recipe evolution process, $EvoRecipes$ replaces an action a_i with another action a_j using $RecipeOn$'s property $alternateAction$. Both a_i and a_j must belong to the same class c'. Detailed categorization of preparatory actions, cooking actions, and post-cooking actions are presented in figure 6.8, figure 6.9, and figure 6.10 respectively. Figure 6.8 shows the sixteen preparatory action categories. Each category has multiple options for replacement. Like for example chopping can be replaced with cutting or dicing. The action substitution operator generates different aromas, tastes, and textures compared to the original recipe.

[6] https://www.allrecipes.com/recipe/272293/fluffy-microwave-scrambled-eggs/

To explain action substitution using an example (noodle bowl[7] (figure 6.6), we have replaced simmer with poach in mutated noodle bowl (figure 6.7).

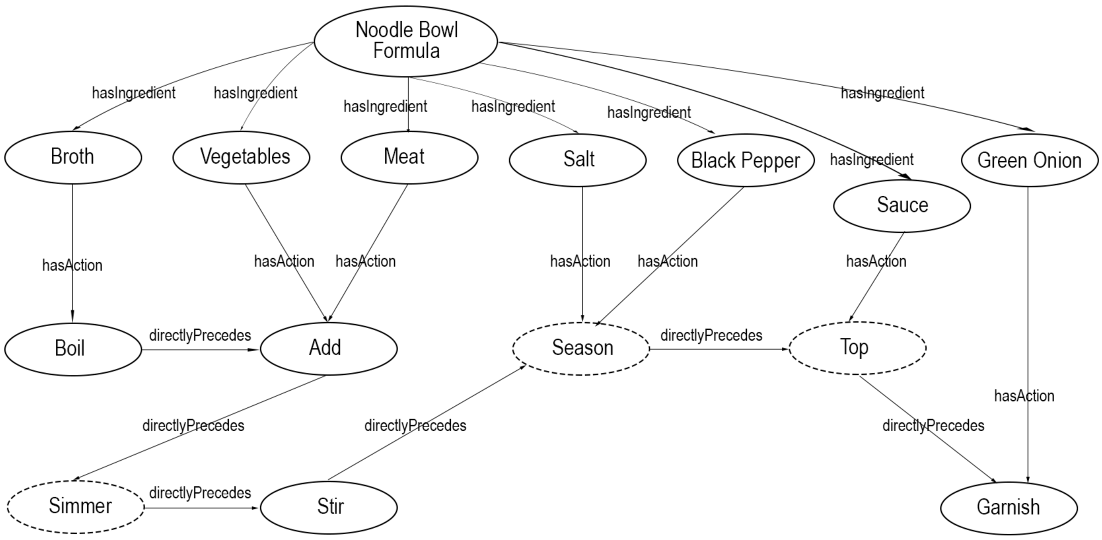

FIGURE 6.6: Noodle Bowl Recipe available at AllRecipes

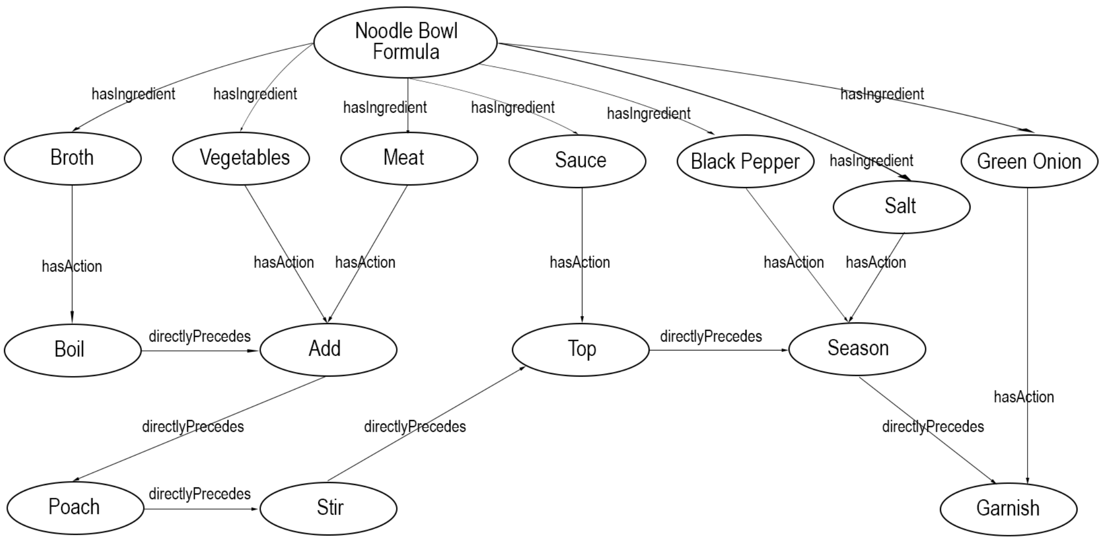

FIGURE 6.7: Mutated (action substitution & action interchanged) Noodle Bowl Formula

6.2.3.3 Action Interchange

Action interchange also introduces novelty in a recipe. Action substitution replaces an action with a potentially similar type of action whereas action interchange neither replaces any action nor adds any new action but just changes

[7] https://www.allrecipes.com/recipe/8493264/noodle-bowl-formula/

FIGURE 6.8: Categorization of Preparatory Actions for Alternative Action Mutation)

the sequence of existing actions. Let's suppose an action a_i $directlyPrecedes$ a_j and a_j $directlyPrecedes$ a_k then after action interchange a_i $directlyPrecedes$ a_k and a_k $directlyPrecedes$ a_j. Both a_j and a_k must belong to the same class. For example, grilling can interchange with toasting or browning as these three actions belong to the same class as shown in figure 6.9. Action interchange for noodle bowl recipe is shown in figure 6.7 in which seasoning is interchanged with topping.

FIGURE 6.9: Categorization of Cooking Actions for Alternative Action (Mutation)

FIGURE 6.10: Categorization of Post-Cooking Actions for Alternative Action (Mutation)

6.2.4 Crossover: Procedure Substitution

A procedure is a subset of a recipe that comprises a subset of related ingredients and actions. Let I be the set of ingredients and A be the set of actions in a recipe. Then a procedure P_j comprises of I_j ingredients and A_j actions such that $I_j \subseteq I$ and $A_j \subseteq A$. The alternative procedure is similar to the crossover operator in evolutionary algorithms. In the alternative procedure, we use the object property $alternativeProcedure$ of $RecipeOn$ ontology to replace a procedure P_i of a recipe r with procedure P_k of another recipe r'. The alternative procedure explores the recipe to a deeper extent and results in a larger change to the recipe compared to the mutation operators.

To explain the crossover operator using an example we have selected two recipes (i.e. "Salmon in Sorrel Sauce"[8] & "Unadon"[9]) as parent recipes. "Salmon in Sorrel Sauce" comprises two procedures. Ingredients and actions involved in the "prepare the sauce" and "prepare the salmon" procedures are shown in figure 6.11 using yellow-colored and green-colored nodes respectively. Similarly, two procedures "Unagi Sauce" and "Rice & Assembly" are shown in figure 6.12 using grey and blue colored nodes respectively for the "Unadon" recipe. Figure 6.13 shows the novel recipe created by replacing the sauce procedure from "prepare the sauce" with "Unagi Sauce" in the "Salmon in Sorrel Sauce" recipe. This alternative procedure results in a change of taste, aroma, and texture of the newly generated recipe.

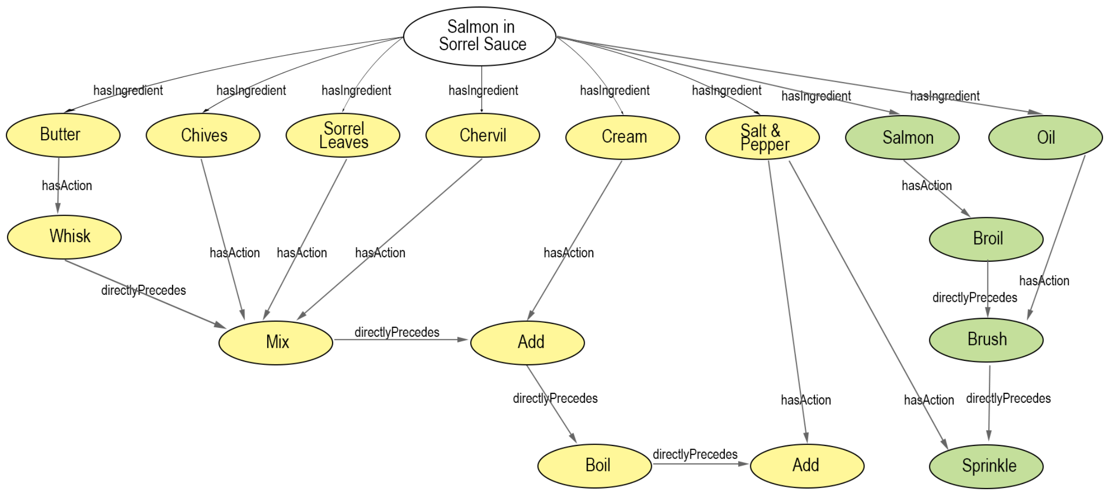

FIGURE 6.11: First Parent Recipe (i.e. Salmon in Sorrel Sauce[8]) Comprising Two Procedures("prepare the Sauce" and "Prepare the Salmon"). Ingredients and Actions Involved in "Prepare the Sauce" are shown using Yellow Colored Nodes while "Prepare the Salmon" is Represented using Green Colored Nodes.

[8] https://food52.com/recipes/4541-salmon-in-sorrel-sauce
[9] https://food52.com/recipes/31455-japanese-eel-rice-bowl-unadon

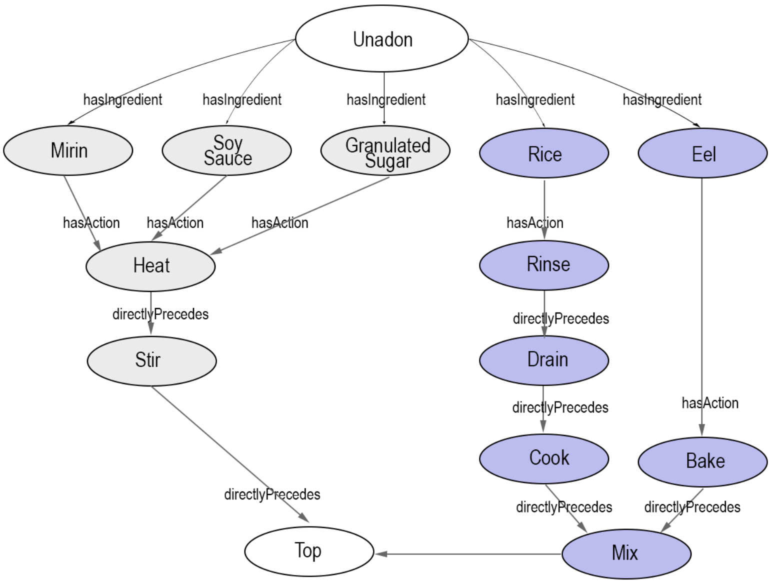

FIGURE 6.12: Second Parent Recipe (i.e. Unadon[9]) Recipe Comprising Two Procedures ("Unagi Sauce" and "Rice & Assembly"). Ingredients and Actions involved in "Unagi Sauce" are shown using Grey Colored Nodes while "Rice & Assembly" is Represented using Blue Colored Nodes.

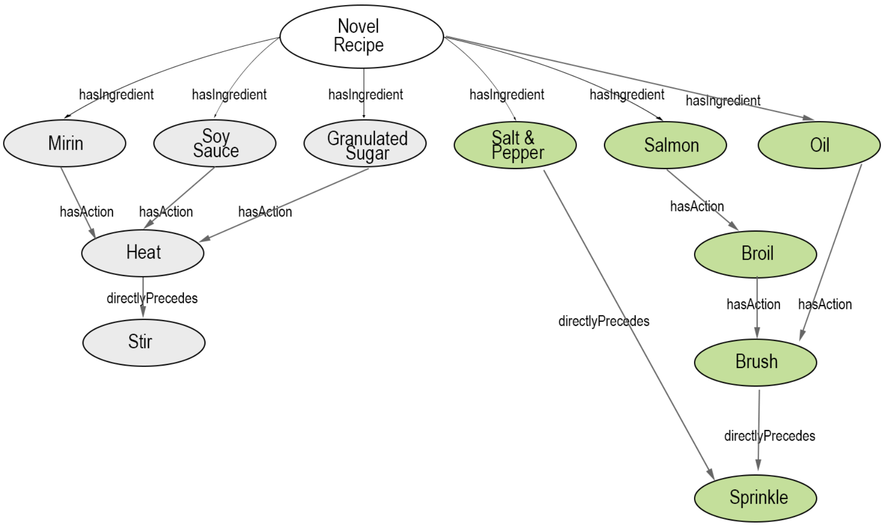

FIGURE 6.13: Novel offspring Recipe Comprising two Procedures ("Unagi Sauce" and "Prepare the Salmon"). Ingredients and Actions involved in "Unagi Sauce" are shown using Grey-Colored Nodes while "Prepare the Salmon" is Represented using Green-Colored Nodes.

6.2.5 Quantitative Evaluation of Recipe

Evaluation of a recipe is more of a subjective task, as it requires humans to assess the quality of a recipe based on taste, texture, aroma, and presentation style. However, to evaluate the alignment of a recipe with the intended goals, quantitative measures are also highly demanded. Each recipe is quantified based on multiple objectives/factors and finally aggregated to assign the fitness value (i.e. score) to a recipe. These quantifiable factors are discussed in detail in the following subsections.

6.2.5.1 Novelty

Novelty (λ) introduces a new combination of flavors, side ingredients, main ingredients, cooking actions, and cooking procedures during evolution that was not present in the initial recipe population. We have classified novelty in a recipe as (i) Novelty in ingredients; (ii) Novelty in actions.

Novelty in ingredients (equation 6.1) is useful for improving the quality of $EvoRecipe$ generated recipes by incorporating unique combinations of ingredients.

$$\lambda_r^I = \frac{1}{N} \sum_{j=1}^{N} \frac{|I_r - I_j|}{|I_r|} \tag{6.1}$$

Here N represents the population size, I_r represents the set of ingredients in an evolved recipe r, and I_j represents the set of ingredients in a recipe j from the initial population. $I_r - I_j$ finds the set of ingredients in recipe r that are not present in recipe j.

Similarly, a novelty in actions (equation 6.2) is useful for improving the quality of evolved recipes by incorporating unique combinations of actions.

$$\lambda_r^A = \frac{1}{N} \sum_{j=1}^{N} \frac{|A_r - A_j|}{|A_r|} \tag{6.2}$$

Here A_r and A_j represent the set of actions in an evolved recipe r and the set of actions in a recipe j (from the initial population), respectively.

$$\lambda_r = \frac{\lambda_r^I + \lambda_r^A}{2} \tag{6.3}$$

The high value of λ_r represents a novel recipe, while a lower value represents a traditional recipe. We have mapped the value of λ_r in equation 6.3 in the range of 0 to 1 by taking the average of λ_r^I and λ_r^A.

6.2.5.2 Simplicity

Simplicity (Θ_r) refers to the easiness involved in preparing a recipe. A simple recipe requires a small number of ingredients, a limited set of preparatory actions, and a small number of sequential steps. We have defined the equation 6.4 to measure the simplicity of a recipe. The high value of Θ_r represents a simple recipe, while the lower value represents a complex recipe.

$$\Theta_r = \frac{1}{|I_r| + 2.|A_r^{prep}| + |A_r^{cook}| + |A_r^{post}| + d_r} \tag{6.4}$$

Here $|A_r^{prep}|$,$|A_r^{cook}|$, and $|A_r^{post}|$ represent the number of preparatory actions, number of cooking actions, and number of post-cooking actions respectively. While d_r refers to the height of a recipe tree. A high value of d_r contributes to a large number of sequential steps. A significant factor that contributes to the recipe's simplicity is the number of preparatory actions. A simple recipe has fewer preparatory actions while a complex recipe requires more preparation before cooking actions. To discourage more preparatory actions in recipes we have assigned a double weight to the reciprocal of A_r^{prep}.

6.2.5.3 Visual Appeal

A visually appealing recipe (ζ_r) simulates the appetite and makes the dining experience more enjoyable and memorable. There are a variety of finishing actions(a subset of post-cooking actions) that contribute to the visual appearance of a recipe. We have proposed equation 6.5, to measure the visual appeal of a recipe.

$$\zeta_r = \frac{|A_r^{fin}|}{|A_r^{post}| + \epsilon} \tag{6.5}$$

Here A_r^{fin} and A_r^{post} represent the number of finishing actions(such as garnishing, sprinkling, topping) and post-cooking actions used in a recipe r. while ϵ is a small positive value that avoids a division by zero in equation 6.5 when a recipe does not have any post-cooking action. ζ_r reflects the ratio of finishing actions to post-cooking actions. A highly appealing recipe would have more concentration on finishing actions than post-cooking actions.

6.2.5.4 Feasibility

A feasible recipe (Υ_r) comprises of preferable ingredients & actions, and avoids undesirable ingredients & actions as specified by the user. This preference can be based on the availability of ingredients, health conditions, and diet consciousness. We have defined the equation 6.6 to measure the feasibility of a recipe.

$$\Upsilon_r = \frac{|I_r^{des}| - |I_r^{undes}|}{2.|I_r|} + \frac{|A_r^{des}| - |A_r^{undes}|}{2.|A_r|} \tag{6.6}$$

Here I_r^{des}, I_r^{undes}, A_r^{des}, and A_r^{undes} represent the number of desired and undesired ingredients and actions respectively. Here Υ_r is the normalized sum of the score that encourages the desired actions and ingredients while punishing the undesirable ingredients and actions.

6.2.5.5 Fitness Function of a Recipe (f)

The fitness of a recipe is compiled based on the score calculated in the above-mentioned factors using equations 6.3, 6.4, 6.5, and 6.6. We have defined equation 6.7 to accumulate the fitness of a recipe that is based on novelty(λ), simplicity(Θ), visual appeal(ζ), and feasibility (Υ). In equation 6.7 we have summed up the factors that contribute to the fitness of a recipe. Here we have multiplied the feasibility factor as it is an important factor while calculating the fitness score of a recipe. Feasibility reflects the recipe's alignment with user preferences. if a recipe is novel, simple, and visually appealing but not feasible then this would result in a minimum fitness value. Moreover, w_1,

w_2, and w_3 are the weighted values that give weightage to the novelty, visual appeal, feasibility, and simplicity based on user choices.

$$f = (w_1.\lambda_r + w_2.\zeta_r + w_3.\Theta_r).\Upsilon_r \tag{6.7}$$

6.2.6 Recipe RDF to Recipe Text using Generative AI

Novel recipes generated from *EvoRecipe* are in RDF format. Recipe RDF comprises classes (i.e. Ingredient, Action, Procedure, Author, AggregateRating, Nutrition, etc) and relationships (i.e. hasAction, hasIngredient, hasProcedure, directlyPrecedes) between the classes or literals. Although the novel recipes are machine-understandable lack human readability. Therefore, a model was needed that could map the recipe facts to human-readable statements. For this purpose, we have used OpenAI GPT API[10] (Davinci model) to create prompts that translate RDF facts to recipe ingredients and instruction statements. Listing 6.1 shows a recipe text generated using OpenAI GPT API against a novel recipe generated by *EvoRecipes*. The prompt corresponding to the generated recipe text was based on RDF format[11].

Listing 6.1: Recipe Text Generated using OpenAI GPT

```
Recipe: Pan-Fried Lamb Chops With Minted
        Pea Salad
Ingredients:
- 4 lamb chops
- 2 tablespoons brown sugar
- Salt to taste
- 1 teaspoon dried basil
- 2 tablespoons grape seed oil
- 1 teaspoon garlic powder
- 2 tablespoons distilled white vinegar
- 1 teaspoon garam masala
- 4 Thai chile peppers, thinly sliced
- 1 cup sugar snap peas
Instructions:
1. Heat the lamb chops in a pan with the
```

[10] https://colab.research.google.com/drive/1yMppDHGs8DODBP4rw9PFieExUv9q4u54?usp=sharing

[11] https://github.com/HajiraJabeen/EvoRecipesOntology/blob/main/prompts/rdf2text

```
   lamb drippings.
2. Coat the lamb chops with the brown
   sugar, salt, and dried basil.
3. Saute the lamb chops in the grape
   seed oil.
4. Add the garlic powder, distilled
   white vinegar, garam masala,
   and Thai chile peppers.
5. Place the lamb chops on a plate.
6. Season the sugar snap peas with salt.
7. Cook the sugar snap peas in a pan.
8. Sprinkle the sugar snap peas with
   garlic powder.
9. Pull the sugar snap peas off the heat.
10. Serve the lamb chops with the
    minted pea salad.
```

6.3 Qualitative Recipe Evaluation

To evaluate our novel recipe generation approach, we also performed a qualitative study on *EvoRecipes*. This evaluation has been carried out in a multitude of dimensions, including contextual evaluation, procedural evaluation, and novelty evaluation. Contextual metrics evaluate (taste, edible, ingredient combination, and action sequence metrics) that fulfills the user preferences. The procedural evaluation focuses on clarity and accuracy to understand the steps involved in a recipe. Recipe validity, complexity, and consistency metrics are covered under this category. Finally, novelty metrics not only evaluate the contribution of evolutionary operators in generating novel recipes but also investigate the quality of generated recipes. The metrics include unusual ingredient combinations, valid action substitution, and valid procedure substitution.

The quality of each recipe is evaluated along three dimensions, i.e., contextual, procedural, and novelty. The metrics used for this multi-dimensional evaluation have been discussed in detail in the following sub-sections.

6.3.1 Contextual Metrics

Contextual recipe evaluation ensures that a recipe is well-aligned with the goals of the intended audience. If some food enthusiasts want to try a new recipe with available ingredients, then a machine-generated recipe should not only be novel but should also be edible and should have a non-conflicting set of ingredients. We have used four contextual metrics to evaluate the quality of a recipe as discussed below.

Taste: The taste of a recipe is based on the appropriate combination of ingredients, their freshness, a suitable sequence of actions, and a balanced use of the flavoring & seasoning ingredients. The taste of a recipe is also based on the user's preference and may vary from person to person. Also, they can set the flavoring ingredients according to the taste they want from the recipe.

Edible: This is a criterion used to find the food items that can be consumed without causing any harm or risk to a person's health. It ensures that the food item is suitable and safe for use. Edible recipes include ingredients that are suitable for consumption. Also, it includes actions (i.e., boiling, baking) that make these ingredients safe to consume by removing harmful bacteria and parasites. In addition to safety, edible recipes should also provide suitable taste, aroma, and texture for a pleasurable eating experience.

Combination of ingredients: This is an important factor in cooking recipes as it influences texture, aroma, and flavor. A suitable and balanced combination of ingredients can make food delicious and visually appealing, and enjoyable. On the other hand, a poor combination can make food non-delicious and unappetizing. The reasons include combining ingredients with strong and conflicting flavors or combining ingredients with similar colors or textures that create a monotone and dull look. Thus the right combination of ingredients can make food delicious, enjoyable, and memorable.

Sequence of actions: Sequence of actions is also important to create well-balanced recipes. Two similar sets of actions but with different sequences can generate food items that would have different textures, flavors, tastes, appearances, and presentations. The sequence of action also affects the cooking time of a recipe, e.g. adding parboiled ingredients helps in reducing the overall cooking time as compared to unboiled ingredients. The sequence of actions affects the interaction of ingredients (e.g. adding lemon juice too early in a recipe can discolor the texture of a vegetable). Some ingredients complement other ingredients while some ingredients conflict with each other. Therefore the compatibility of ingredients is also based on the right sequence of actions. Lastly, the food safety perspective is also considered in the right sequence of actions. For example, it is important to cook meat thoroughly to kill harmful bacteria. Therefore beef should be added earlier in a recipe than other ingredients to ensure that it imposes no health risks. Understanding the importance of a sequence of actions can help to generate food items that are safe, tasty, and visually appealing.

6.3.2 Procedural Metrics

Procedural metrics evaluate the clarity of steps involved in a recipe. It also assures that the steps are easy to understand and follow. These metrics are discussed in detail as follows.

Validity: Validity ensures that a recipe is comprised of a clear, unambiguous, and concise set of instructions along with an appropriate sequence of actions. Moreover, it has a detailed list of ingredients, quantities of ingredients, and reasonable steps related to preparation, cooking, and post-cooking actions.

Complexity: Complexity refers to the level of difficulty in following the step-by-step instructions of a recipe. The goal of this metric is to ensure that the recipe can be followed by the reader with a reasonable level of cooking experience and skill to create a food item that is consistent with the intended outcome. Professional cooks often conduct complexity evaluations to ensure that the recipes are well-written and understandable. Complexity arises due to inconsistencies, gaps, and ambiguities that lead to confusion during preparation. Overall a recipe should be easy to understand and follow.

Usability: Usability refers to the ease with which a recipe can be prepared by cooks with varying levels of skills and experience. It is based on the factors like availability of ingredients, accessibility of equipment, and the overall user-friendliness of the recipe. A recipe with poor usability requires specialized equipment, ingredients with low availability, and skills that are beyond the skill set of many cooks. Usability ensures that the recipes can be successfully prepared by a wide range of cooks.

6.3.3 Novelty Metrics

Novelty evaluation determines whether a recipe offers a unique or creative approach to a food item or floats an entirely new idea that hasn't been explored previously. It measures the originality or creativity of *EvoRecipe* generated recipes. Practically, novelty evaluation for computational creative culinary recipes involves comparing newly generated recipes with a large corpus of cooking recipes. An alternative to this approach is to get novel recipes assessed by human tasters. This evaluation helps to identify novel and unique recipe ideas that may have been unexplored or overlooked previously. We suggest three novelty evaluation parameters to asses the originality or uniqueness of a recipe as discussed below.

Unusual ingredient combination: The unusual combination of ingredients metric helps credit recipes that have a rare combination of ingredients (i.e. not available commonly in human-generated recipes). A rare ingredient combination will lead to recipes that would be very different in their taste, aroma, visual appearance, and texture. Hence leading to more novel recipes.

Rare Action Substitution: A recipe comprises preparatory actions, cooking actions, and post-cooking actions. Alternative actions consider how unique or unusual preparatory actions and cooking actions have been used in a recipe compared to other similar recipes. If a machine-generated recipe applies more unique cooking actions on available ingredients then it would be considered more novel compared to a recipe that applies more traditional actions on ingredients.

TABLE 6.1: Recipe Evaluation Comparing Human-generated Recipe with EvoRecipe Generated Recipe.

Evaluation Metric	Human-generated Recipe	EvoRecipe generated Recipe	Both of them	None
Taste	09	**10**	03	02
Edible	04	07	**14**	-
Appropriate Combination of Ingredient	**13**	07	04	01
Sequence of Action	08	06	**09**	02
Validity	05	04	**14**	01
Complexity	**13**	10	-	01
Usability	09	**13**	03	-
Unusual Ingredient Combination	06	**11**	01	06
Rare Action Substitution	06	**08**	03	07
Rare Procedure Substitution	**09**	**09**	01	05
Readable	09	**12**	04	-

Rare Procedure Substitution: A procedure or sub-recipe (component recipe within a main recipe) is a subset of a recipe that involves ingredients and corresponding cooking actions. This metric assesses the novelty of a recipe based on the usage of unique or new procedures within a recipe. If a machine-generated recipe introduces a unique sauce procedure or replaces a traditional sauce sub-recipe with an unusual sauce sub-recipe then it is considered more novel compared to a recipe that replaces a procedure with another commonly used procedure. This evaluation technique helps to find more unique sub-recipes that have not been used before with similar recipes and discourages finding commonly used sub-procedures with similar recipes.

6.3.4 Survey and Analysis

To evaluate the qualitative metric of the *EvoRecipes* approach, we have conducted a qualitative study. The study aims to investigate the contribution of evolutionary operators (mutation & crossover) in generating novel recipes and the contribution of generative AI (OpenAI GPT) to convert those recipes into human-readable text. In addition, this study is helpful for evaluating the quality of *EvoRecipe* generated recipes.

To conduct the study we have used two recipes. The first recipe was chosen randomly from *RecipeKG* that reflects an original human-generated recipe from a recipe website, while the second recipe was generated using the *EvoRecipe* framework.

The quality of a culinary recipe is a subjective matter therefore we have designed a survey (using Google Forms) to ask participants to assess the quality of the floated recipes. The survey comprises 11 multiple choice questions (that is, at least one against each evaluation metric as described in sections 6.3.1,6.3.2, and 6.3.3) to be filled out by the participants. Each question follows a template (i.e. Which of the two recipes is/has <evaluation-metric>? e.g., Which of the two recipes is more usable?) with four options against each question (Recipe A, Recipe B, Both of them, None). The survey questions are listed below.

1. Which of the two recipes is edible?
2. Which of the two recipes is valid?
3. Which of the two recipes is complex?
4. Which of the two recipes has an appropriate ingredient combination?
5. Which of the two recipes has unusual ingredient combinations?
6. Which of the two recipes has the appropriate sequence of instructions?
7. Which of the two recipes has rare actions/instructions?
8. Which of the two recipes has a rare procedure?
9. Which of the two recipes is more tasty?
10. Which of the two recipes is prepared easily?
11. Which of the two recipes is more readable?

The survey is filled out by 24 users and the results are presented in table 6.1. The results show that the *EvoRecipe* generated recipes are more usable (13 users recommended *EvoRecipe* generated recipes whereas 9 users recommended human-generated recipes), readable (12 users recommended *EvoRecipe* generated recipes whereas 9 users recommended human-generated recipes), having unusual ingredient combination (11 users recommended *EvoRecipe* generated recipes whereas 6 users recommended human-generated recipes), rare action substitution (8 users recommended *EvoRecipe* generated recipes whereas 6 users recommended human-generated recipes), simple (10 users rated *EvoRecipe* generated recipes as complex recipe whereas 13 users rated human-generated recipes as a complex recipe), and tasty (10 users recommended *EvoRecipe* generated recipes whereas 9 users recommended human-generated recipes) than human-generated recipe. Moreover, both recipes are valid (14 users recommended *EvoRecipe* generated recipes and human-generated recipes) and have an appropriate sequence of actions (09 users recommended *EvoRecipe* generated recipes and human-generated recipes). *EvoRecipe* performed better in 07 qualitative metrics (Taste, Edible, Usability, simplicity, Unusual ingredient combination, rare action substitution, and readable) equal to 1 which is rare procedure substitution as compared to human-generated recipes.

6.4 Result and Discussion

6.4.1 Experimental Setup

We have performed experiments on Google Colab using Python 3 Google Compute Engine with 12 GB Ram and 107 GB secondary storage. We have selected an initial population of 100 recipes from RecipeKG as a sample to evolve the recipes. This population comprises lamb recipes, poultry recipes, rice recipes, and Asian noodle recipes that has lamb, poultry, rice, and Asian noodles as the main ingredient respectively.

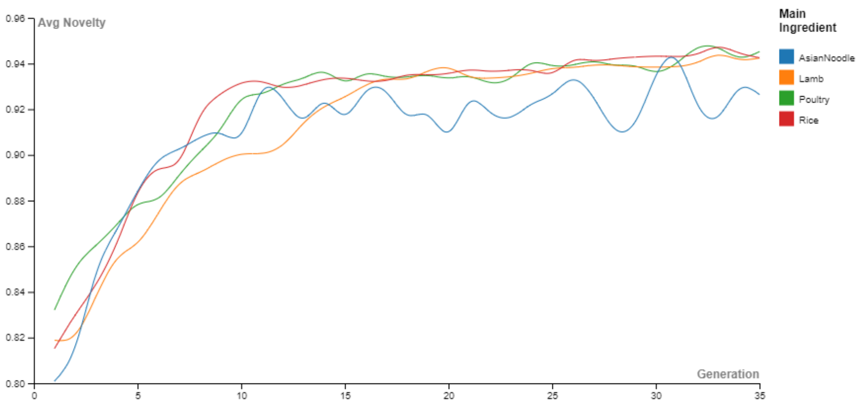

FIGURE 6.14: Average Novelty Over 10 Runs of *EvoRecipes*

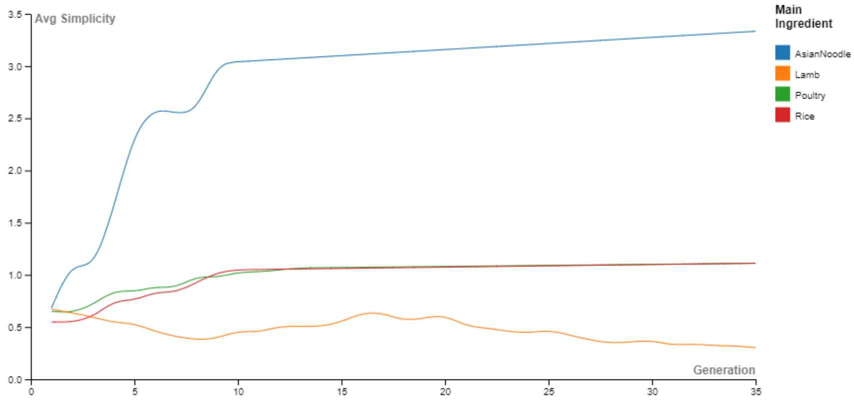

FIGURE 6.15: Average Simplicity Over 10 Runs of *EvoRecipes*

To evolve the recipes we have implemented Genetic Algorithm [12] in Python. Concerning parameter settings single mutation of either ingredient or action node is performed for each recipe. Parent recipes for crossover are selected using tournament selection with a crossover rate of 0.5. The roulette wheel has been used for the next generation's population selection while the stopping criteria is 35 generations. These parameter values were taken from autochef (Jabeen et al., 2020).

6.4.2 Experiments

EvoRecipes has been executed 10 times to get the average metrics values (novelty, simplicity, visual appeal, and fitness) as shown in figure 6.14, figure 6.15, figure 6.16, and figure 6.17 respectively. These figures presented the results of the quantitative evaluation of recipes generated through *EvoRecipe*. Figure 6.14 shows the continuous improvement of novelty for all four types of recipes. Mutation dominates crossover in novelty improvement due to the reason that crossover just incorporates new ingredients/actions that are available in other recipes of the current population, while mutation incorporates new ingredients and actions from *RecipeOn* ontology that are not even available in the population. Hence mutation generates unusual/unique ingredients and action combinations. The value of novelty (λ_r) is calculated using equation 6.3 and represents an unusual ingredient combination and varying actions in a novel recipe

[12] https://github.com/HajiraJabeen/EvoRecipesOntology/blob/main/EvoRecipes/EvoRecipes.py

compared to the set of ingredients and actions in the initial population. Figure 6.14 shows that Asian noodles have slightly less novelty value compared to lamb, poultry, and rice recipes. This is because ingredients involved in Asian noodles have limited alternative options and restrict to the incorporation of unusual ingredient combinations.

Figure 6.15 shows that Asian noodles recipes are more simple to make as they involve a lesser number of ingredients and few preparatory actions, while lamb recipes are more complex as it generally involves more ingredients and actions. However, poultry and rice recipes are simpler than lamb recipes while complex than Asian noodles recipes. The simplicity (Θ_r) of a recipe is calculated using equation 6.4 and is generally an overlooked factor. A simpler recipe is not only easy to remember (due to a limited number of ingredients and actions) but also requires less amount of effort to prepare a food item.

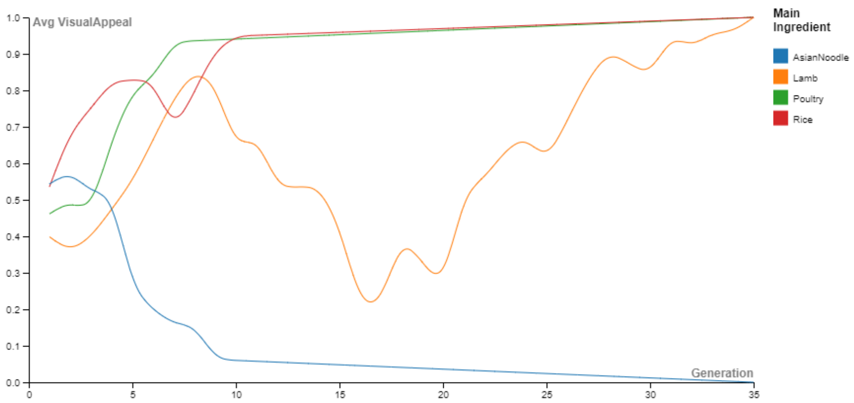

FIGURE 6.16: Average Visual Appeal Over 10 Runs of *EvoRecipes*

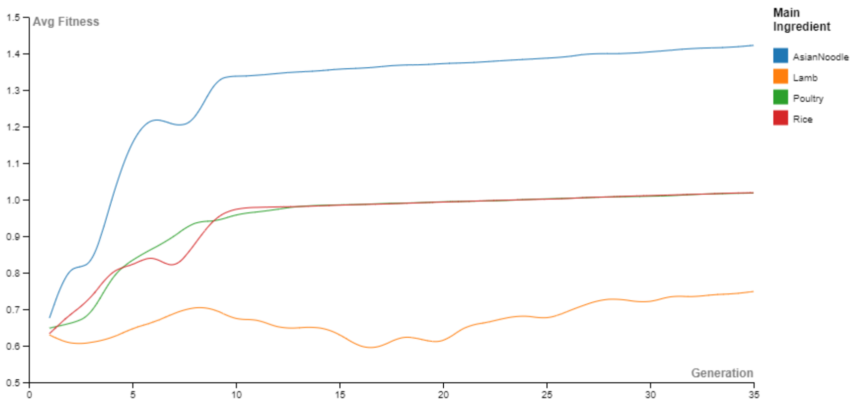

FIGURE 6.17: Average Fitness Value Over 10 Runs of *EvoRecipes*

Figure 6.16 shows that poultry, rice, and lamb recipes are more visually appealing compared to Asian noodle recipes. Poultry, rice, and lamb recipes have more finishing actions (i.e. garnishing, topping, layering, drizzling) that help to improve their visual appeal. While Asian noodles have a low average visual appeal score because they have limited finishing actions. Figure 6.16 shows the Visual Appeal (ζ_r) score calculated through equation 6.5, for the newly generated recipes through *EvoRecipe*. Poultry, Rice, and Lamb recipes have more visual appeal scores than noodle recipes as they have more finishing actions that make a recipe visually appealing.

The average fitness score (f) of novel poultry, lamb, rice, and Asian noodle recipes (generated through *EvoRecipe*) is shown in figure 6.17. f has been calculated using equation 6.7. Asian noodles have high fitness values compared to poultry, rice, and lamb recipes. This is due to the reason that Asian noodle recipes are simpler than counterpart recipes while in terms of novelty, Asian noodle recipes are closer to the fitness value of lamb, poultry, and rice. While poultry and rice recipes have similar fitness values across many generations as they have close values of novelty, simplicity, and visual appeal across many generations. Lamb recipes show less fitness score (f) value as they are complex and have less visual appeal as compared to poultry, rice, and noodle recipes. Results in Figures 6.14, 6.15, 6.16, and figure 6.17 proves that recipes generated using *EvoRecipes* are novel, simple, visually appealing, and valid.

6.4.3 Comparison with Recipe Evolution Techniques

EvoChef (Jabeen et al., 2019), and AutoChef (Jabeen et al., 2020) have used genetic algorithms and genetic programming respectively to evolve the recipes. EvoChef has only covered the limited potato recipes and has lacked in defining fitness functions for recipe evaluation. They have evaluated recipes manually by human experts. While AutoChef has used genetic programming to evolve the recipes. They have tried limited food replacement options and a few action replacement options. They are also not taking care of the sequencing of instructions and actions while performing recipe evolution. Moreover, they are not using context-aware machine-understandable recipes. However, *EvoRecipes* is based on the schema of *RecipeOn* ontology that encodes rich information related to a recipe. It represents the recipe as a process and includes ingredients, actions, nutrition, and sequence of actions. Furthermore, *EvoRecipes* evolves recipes by applying ingredient substitution, action substitution, action interchange, and procedure substitution while using the semantically rich schema of *RecipeOn* ontology. The ontology provides support to *EvoRecipe* invalid substitution of ingredients and actions using rich recipe knowledge, class hierarchy detail of ingredients & actions, and sequence of actions. Also, *EvoRecipe* ensures the valid procedure substitution while evolving the recipe as it uses the alternative procedure rules defined by the *RecipeOn* ontology. *EvoRecipe* ensures the quantitative evaluation of the recipe through its proposed fitness function and the qualitative evaluation of the recipe through its proposed multi-perspective metrics.

6.5 Summary

In this chapter, we have proposed the recipe evolution framework *EvoRecipes* that randomly selects the initial population from the *RecipeKG* knowledge graph. *EvoRecipes* uses *RecipeOn* ontology to mutate the recipes using three different mutation operators (i.e. ingredient substitution, action substitution, and action interchange). Moreover, in procedure substitution, it interchanges the procedures of two selected recipes (through tournament selection) to perform the crossover operation. Finally, for the newly created recipes, we have generated the human-readable recipe text using OpenAI GPT API (Davinci model). Also, we have proposed quantitative metrics that contribute to the fitness evaluation of a recipe. Furthermore, we have carried out a qualitative study using the survey to evaluate the subjective parameters (like taste, edible, complexity, usability, novelty, etc.) of a recipe. Currently, *EvoRecipes* does not take care of the nutritional details while evolving recipes.

Chapter 7

Conclusion & Future Directions

In this thesis, we have developed an intelligent system that uses computational creativity to generate novel context-aware recipes aligned with user preferences. The system comprises three main components ($RecipeOn$, $RecipeKG$, and $EvoRecipes$). $RecipeOn$ is an Ontology for representing cooking recipes. It follows a modular approach that comprises $Core - Recipe$, $Ingredients$, $Actions$, $Nutrition$, and $Procedure$ modules. $RecipeOn$ successfully maps any culinary recipe as a process and is capable to answer the competency questions designed for intended users (sick persons, diet-conscious people, and food enthusiasts). $RecipeOn$ helps in customized recipe generation and is designed following NeOn methodology while using the classes and properties from existing ontologies (i.e., owl, Seq, schema, and Qudt). Moreover, $RecipeOn$ can be applied to our specifically designed AI-based use case of novel recipe generation, achievable through alternative ingredients, instructions, and procedures. $RecipeOn$ is also consistent with ontology quality parameters. $RecipeOn$ and its documentation is available online and hence it is adaptable and reusable. $RecipeOn$ makes it easy for the users to follow the recipe steps using ingredient-actions, and action-action relationships.

Moreover, we have also compiled a 0.8M-Recipes dataset that comprises 50+ attributes. 0.8M-Recipes dataset has been compiled after data extraction, data consolidation, data preprocessing, and data transformation phases. On top of 0.8M-Recipes, we have generated a large-scale semantic-aware recipe knowledge graph $RecipeKG$ that increases machine-understandably of the recipe process. $RecipeKG$ contains 0.8 million recipe nodes, 8.9 million ingredient nodes, 10.8 million action nodes, 1 million nutrient nodes, and 209 million facts. $RecipeKG$ helps to create machine-generated recipes using appropriate alternative actions and ingredients. $RecipeKG$ provides ingredient-ingredient,

ingredient-action, and action-action relationships using the concepts and properties from *RecipeOn* ontology. Moreover, we have presented use cases for recipe profiling, and a recipe for sick & diet-conscious persons is presented along with listings for competency questions.

Finally, we have proposed the recipe evolution framework *EvoRecipes* that randomly selects the initial population from *RecipeKG* knowledge graph. *EvoRecipes* uses *RecipeOn* ontology to mutate the recipes using three different mutation operators (i.e. ingredient substitution, action substitution, and action interchange). Moreover, in procedure substitution, it interchanges the procedures of two selected recipes (through tournament selection) to perform the crossover operation. Finally, for the newly created recipes, we have generated the human-readable recipe text using OpenAI GPT API (Davinci model). Also, we have proposed quantitative metrics that contribute to the fitness evaluation of a recipe. Furthermore, we have carried out a qualitative study using the survey to evaluate the subjective parameters (like taste, edible, complexity, usability, novelty, etc.) of a recipe. Currently, *EvoRecipes* does not take care of the nutritional details while evolving recipes.

In the future, the 0.8M-Recipes dataset would be extended to include more recipes for different cuisines. We will also extend the recipe knowledge graph *RecipeKG* and add more instances and facts. We also aim to generalize the *RecipeOn* ontology to easily adapt to other processes like construction and industrial manufacturing. Furthermore, we will extend evaluation metrics to tune the parameters that are involved in recipe evolution. Also, we aim to explore the distributed in-memory computational frameworks (like Apache Spark, and Apache Flink) to speed up the recipe evolution process and to increase the scalability of the *EvoRecipe* framework.

Bibliography

I. Agarwal. Intelligent refrigerator. In *Intelligent Systems, Technologies and Applications*, pages 241–258. Springer, 2020.

R. Agarwal, D. Jachowski, and D. Shengxi. Robochef: automatic recipe generation. *Tech. Np*, 2009.

Y.-Y. Ahn and S. Ahnert. The flavor network. *Leonardo*, 46(3):272–273, 2013.

Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási. Flavor network and the principles of food pairing. *Scientific reports*, 1(1):1–7, 2011.

M. Al-Razgan, S. Tallab, and T. Alfakih. Exploring the food pairing hypothesis in saudi cuisine using genetic algorithm. *Mathematical Problems in Engineering*, 2021:1–16, 2021.

M. Amith, J. Wang, G. Xiong, K. Roberts, and C. Tao. A health consumer ontology of fast food information. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1714–1719. IEEE, 2020.

M. Amith, C. Onye, T. Ledoux, G. Xiong, and C. Tao. The ontology of fast food facts: conceptualization of nutritional fast food data for consumers and semantic web applications. *BMC medical informatics and decision making*, 21 (7):1–16, 2021.

W. Antô, J. R. Bezerra, L. F. W. Góes, F. M. F. Ferreira, et al. Creative culinary recipe generation based on statistical language models. *IEEE Access*, 8: 146263–146283, 2020.

T. Arffa, R. Lim, and J. Rachleff. Learning to cook: An exploration of recipe data, 2016.

M. T. Artese, G. Ciocca, and I. Gagliardi. Analysis of traditional italian food recipes: Experiments and results. In *International Conference on Pattern Recognition*, pages 677–690. Springer, 2021.

T. Bailoni, M. Dragoni, C. Eccher, M. Guerini, and R. Maimone. Healthy lifestyle support: the perkapp ontology. In *OWL: Experiences and Directions–Reasoner Evaluation*, pages 15–23. Springer, 2016.

F. Batista, J. P. Pardal, P. V. N. Mamede, and R. Ribeiro. Ontology construction: cooking domain. *Artificial Intelligence: Methodology, Systems, and Applications*, 41(1):30, 2006.

D. Batra, N. Diwan, U. Upadhyay, J. S. Kalra, T. Sharma, A. K. Sharma, D. Khanna, J. S. Marwah, S. Kalathil, N. Singh, et al. Recipedb: A resource for exploring recipes. *Database*, 2020, 2020.

BBC. Bbc - ontologies - food ontology, 2014. URL https://www.bbc.co.uk/ontologies/fo. Accessed: 2021-06-01.

O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar. Menu-match: Restaurant-specific food logging from images. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 844–851. IEEE, 2015.

M. Bień, M. Gilski, M. Maciejewska, W. Taisner, D. Wisniewski, and A. Lawrynowicz. Recipenlg: A cooking recipes dataset for semi-structured text generation. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 22–28, 2020.

S. Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, 2006.

M. Bolaños, A. Ferrà, and P. Radeva. Food ingredients recognition through multi-label learning. In *International Conference on Image Analysis and Processing*, pages 394–402. Springer, 2017.

G. Bordea, J. Nikiema, R. Griffier, T. Hamon, and F. Mougin. Fideo: Food interactions with drugs evidence ontology. In *11th International Conference on Biomedical Ontologies*, 2020.

L. Bossard, M. Guillaumin, and L. Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014.

S. Camarena. Artificial intelligence in the design of the transitions to sustainable food systems. *Journal of Cleaner Production*, 271:122574, 2020.

P. Castellano-Escuder, R. González-Domínguez, D. S. Wishart, C. Andrés-Lacueva, and A. Sánchez-Pla. Fobi: An ontology to represent food intake data and associate it with metabolomic data. *Database*, 20, 2020.

D. Çelik. Foodwiki: Ontology-driven mobile safe food consumption system. *The scientific World journal*, march-15, 2015.

L. D. Chambers. *Practical Handbook of Genetic Algorithms: Complex Coding Systems, Volume III*, volume 3. CRC press, 2019.

J. Chen and C.-W. Ngo. Deep-based ingredient recognition for cooking recipe retrieval. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 32–41, 2016.

J. Chen, B. Zhu, C.-W. Ngo, T.-S. Chua, and Y.-G. Jiang. A study of multi-task and region-wise deep learning for food ingredient recognition. *IEEE Transactions on Image Processing*, 30:1514–1526, 2020.

J.-j. Chen, C.-W. Ngo, and T.-S. Chua. Cross-modal recipe retrieval with rich food attributes. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1771–1779, 2017.

J.-J. Chen, C.-W. Ngo, F.-L. Feng, and T.-S. Chua. Deep understanding of cooking procedure for cross-modal recipe retrieval. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1020–1028, 2018.

M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang. Pfid: Pittsburgh fast-food image dataset. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 289–292. IEEE, 2009.

J. Clunis. Designing an ontology for managing the diets of hypertensive individuals. *International Journal on Digital Libraries*, 20(3):269–284, 2019.

M. De Klepper. Food pairing theory: A european fad. *Gastronomica: The Journal of Food and Culture*, 11(4):55–58, 2011.

J. Deng, P. Olivier, J. Andres, K. Ellis, R. Wee, and F. Floyd Mueller. Logic bonbon: exploring food as computational artifact. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2022.

D. M. Dooley, E. J. Griffiths, G. S. Gosal, P. L. Buttigieg, R. Hoehndorf, M. C. Lange, L. M. Schriml, F. S. Brinkman, and W. W. Hsiao. Foodon: a harmonized food ontology to increase global food traceability, quality control and data integration. *npj Science of Food*, 2(1):1–10, 2018.

M. Dragoni, T. Bailoni, R. Maimone, and C. Eccher. Helis: An ontology for supporting healthy lifestyles. In *International semantic web conference*, pages 53–69. Springer, 2018.

C. Draschner, J. Lehmann, and H. Jabeen. Smart chef: Evolving recipes. *EVO*, Leipzig*, 2019.

T. Eftimov, G. Ispirova, D. Potočnik, N. Ogrinc, and B. Seljak. Iso-food ontology: A formal representation of the knowledge within the domain of isotopes for food science. *Food Chemistry*, 277, 10 2018. doi: 10.1016/j.foodchem.2018.10.118.

T. Eftimov, G. Ispirova, D. Potočnik, N. Ogrinc, and B. K. Seljak. Iso-food ontology: A formal representation of the knowledge within the domain of isotopes for food science. *Food chemistry*, 277:382–390, 2019.

D. Elsweiler, C. Trattner, and M. Harvey. Exploiting food choice biases for healthier recipe recommendation. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval*, pages 575–584, 2017.

P. Engisch. Recipes and culinary creativity. *HUMANA. MENTE Journal of Philosophical Studies*, 13(38):61–82, 2020.

Foodwiki. Foodwiki, 2014. URL https://www.takeaway.com/foodwiki/. Accessed: 2021-06-01.

D. Garijo. Widoco: a wizard for documenting ontologies. In *International Semantic Web Conference*, pages 94–102. Springer, 2017.

M. Gim, D. Park, M. Spranger, K. Maruyama, and J. Kang. Recipebowl: A cooking recommender for ingredients and recipes using set transformer. *IEEE Access*, 9:143623–143633, 2021.

M. Gim, D. Choi, K. Maruyama, J. Choi, H. Kim, D. Park, and J. Kang. Recipemind: Guiding ingredient choices from food pairing to recipe completion using cascaded set transformer. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3092–3102, 2022.

J. Harashima, M. Ariga, K. Murata, and M. Ioki. A large-scale recipe and meal data collection as infrastructure for food research. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2455–2459, 2016.

J. Harashima, Y. Someya, and Y. Kikuta. Cookpad image dataset: An image collection as infrastructure for food research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1229–1232, 2017.

S. Haussmann, O. Seneviratne, Y. Chen, Y. Ne'eman, J. Codella, C.-H. Chen, D. L. McGuinness, and M. J. Zaki. Foodkg: a semantics-driven knowledge graph for food recommendation. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*, pages 146–162. Springer, 2019.

H. Jabeen, N. Tahara, and J. Lehmann. Evochef: show me what to cook! artificial evolution of culinary arts. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*, pages 156–172. Springer, 2019.

H. Jabeen, J. Weinz, and J. Lehmann. Autochef: Automated generation of cooking recipes. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–7. IEEE, 2020.

A. Jain, R. NK, and G. Bagler. Analysis of food pairing in regional cuisines of india. *PloS one*, 10(10):e0139539, 2015.

K. Jain. 6000+ Indian Food Recipe Dataset. https://data.mendeley.com/datasets/xsphgmmh7b/1, 2020. [Online; accessed 27-June-2021].

A. Jimenez-Mavillard and J. L. Suarez. A computational approach for creativity assessment of culinary products: the case of elbulli. *AI & SOCIETY*, 37(1): 331–353, 2022.

A. F. U. R. Khilji, R. Manna, S. R. Laskar, P. Pakray, D. Das, S. Bandyopadhyay, and A. Gelbukh. Cookingqa: answering questions and recommending recipes based on ingredients. *Arabian Journal for Science and Engineering*, 46(4): 3701–3712, 2021.

M. Kolchin and D. Zamula. Food product ontology: Initial implementation of a vocabulary for describing food products. In *Proceeding of the 14th Conference of Open Innovations Association FRUCT, Helsinki, Finland*, pages 11–15, 2013.

M. Kolchin and D. Zamula. Food ontology: Ontology for describing food products. In *13th Conference of fruct association*, pages 194–196, 2020.

K. Kusu, H.-I. Choi, T. Kambara, T. Kinoshita, T. Shioi, and K. Hatano. Searching cooking recipes by focusing on common ingredients. In *Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services*, pages 95–101, 2017.

D. Lawo, L. Böhm, and G. Stevens. Veganaizer: Ai-assisted ingredient substitution, 2020.

A. Ławrynowicz. Creative ai: A new avenue for the semantic web? *Semantic Web*, 11(1):69–78, 2020.

N. Le Guillarme, M. Hedde, and W. Thuiller. Building a trophic knowledge graph to support soil food web reconstruction. In *S4BioDiv 2021: 3rd International Workshop on Semantics for Biodiversity and JOWO 2021: Episode VII The Bolzano Summer of Knowledge*, 2021.

B. M. Lee. Food ontology model for a healthcare service. *Journal of the Korea Industrial Information Systems Research*, 17(6):31–40, 2012.

Z. Lei, A. U. Haq, A. Zeb, M. Suzauddola, and D. Zhang. Is the suggested food your desired?: Multi-modal recipe recommendation with demand-based knowledge graph. *Expert Systems with Applications*, 186:115708, 2021.

P. Lertkrai, N. Kaewboonma, and J. Lertkrai. Developing ontology of food and nutrition for thai pre-school. In *AIP Conference Proceedings*, page 020072. AIP Publishing LLC, 2018.

D. Li and M. J. Zaki. Reciptor: An effective pretrained model for recipe representation learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1719–1727, 2020.

R. Loughran and M. O'Neill. Application domains considered in computational creativity. In *ICCC*, pages 197–204, 2017.

F. Lu, N. Kochura, J. Janakiraman, and S. M. Cox. Modification of food recipes based on geographic origin of produce, 2020. US Patent App. 16/204,727.

D. P. Madalli, U. Chatterjee, and B. Dutta. An analytical approach to building a core ontology for food. *Journal of Documentation*, 2017.

B. P. Majumder, S. Li, J. Ni, and J. McAuley. Generating personalized recipes from historical user preferences. *arXiv preprint arXiv:1909.00105*, 2019.

R. Manna, P. Pakray, S. Banerjee, D. Das, and A. Gelbukh. Cookingqa: A question answering system based on cooking ontology. In *Mexican International Conference on Artificial Intelligence*, pages 67–78. Springer, 2016.

R. Manna, D. Das, and A. Gelbukh. Question-answering and recommendation system on cooking recipes. *Computación y Sistemas*, 25(1), 2021.

C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

J. Marin, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):187–203, 2019.

W. Min, C. Liu, L. Xu, and S. Jiang. Applications of knowledge graphs for food science and industry. *Patterns*, 3(5):100484, 2022.

M. Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

R. Mizoguchi. Tutorial on ontological engineering part 2: Ontology development, tools and languages. *New Generation Computing*, 22(1):61–96, 2004.

R. G. Morris, S. H. Burton, P. M. Bodily, and D. Ventura. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *ICCC*, pages 119–125. Citeseer, 2012.

M. Musen, N. Noy, N. Shah, P. Whetzel, C. Chute, M. Storey, and B. Smith. The national center for biomedical ontology. *Journal of the American Medical Informatics Association : JAMIA*, 19 2:190–5, 2012a.

M. A. Musen, N. F. Noy, N. H. Shah, P. L. Whetzel, C. G. Chute, M.-A. Story, B. Smith, and N. team. The national center for biomedical ontology. *Journal of the American Medical Informatics Association*, 19(2):190–195, 2012b.

W. Nadee and S. Unankard. Alternative-ingredient recommendation based on correlation weight for thai recipes. In *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, pages 14–17. IEEE, 2021.

J. Naik and V. Polamreddi. Cuisine classification and recipe generation. *Online: http://cs229. stanford. edu/proj2015/233 report. pdf*, 2015.

M. Ozgen. Tagging and action graph generation for recipes. Master's thesis, Hacettepe University, 2019.

V. Pallagani, P. Ramamurthy, V. Khandelwal, R. Venkataramanan, K. Lakkaraju, S. N. Aakur, and B. Srivastava. A rich recipe representation as plan to support expressive multi modal queries on recipe content and preparation process. *arXiv preprint arXiv:2203.17109*, 2022.

Y. Pan, Q. Xu, and Y. Li. Food recipe alternation and generation with natural language processing techniques. In *2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*, pages 94–97. IEEE, 2020.

D. Park, K. Kim, S. Kim, M. Spranger, and J. Kang. Flavorgraph: a large-scale food-chemical graph for generating food representations and recommending food pairings. *Scientific reports*, 11(1):1–13, 2021.

D. Pereira, A. Bozzato, P. Dario, and G. Ciuti. Towards foodservice robotics: a taxonomy of actions of foodservice workers and a critical review of supportive technology. *IEEE Transactions on Automation Science and Engineering*, 19(3): 1820–1858, 2022.

A. G. Pérez, M. D. F. Baonza, and B. Villazón. Neon methodology for building ontology networks: Ontology specification. *Methodology*, pages 1–18, 2008.

F. Pinel and L. R. Varshney. Computational creativity for culinary recipes. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 439–442. ACM, 2014.

T. Pizzuti, G. Mirabelli, M. A. Sanz-Bobi, and F. Goméz-Gonzaléz. Food track & trace ontology for helping the food traceability control. *Journal of Food Engineering*, 120:17–30, 2014.

G. Popovski, B. K. Seljak, and T. Eftimov. Foodbase corpus: a new resource of annotated food entities. *Database*, 2019, 2019.

M. Poveda-Villalón, A. Gómez-Pérez, and M. C. Suárez-Figueroa. Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2):7–34, 2014.

M. Qi, Y. Neeman, F. Eckhardt, and K. Blissett. Whattomake: A semantic web application for recipe recommendation. *Rensselaer Polytechnic Institute*, 2018.

L. Qin, Z. Hao, and L. Zhao. Food safety knowledge graph and question answering system. In *Proceedings of the 2019 7th International Conference on Information Technology: IoT and Smart City*, pages 559–564, 2019.

L. Qin, Z. Hao, and L. Yang. Question answering system based on food spot-check knowledge graph. In *Proceedings of 2020 the 6th international conference on computing and data engineering*, pages 168–172, 2020.

A. F. U. Rahman Khilji, R. Manna, S. Rahman Laskar, P. Pakray, D. Das, S. Bandyopadhyay, and A. Gelbukh. Question classification and answer extraction for developing a cooking qa system. *Computación y Sistemas*, 24(2): 927–933, 2020.

K. S. Reddy, K. C. Reddy, P. Chennareddy, and N. Jahan. Diet recommendation system based on vitamin intake. *Advances in Computational Sciences and Technology*, 16(1):35–43, 2023.

R. Ribeiro, F. Batista, J. P. Pardal, N. J. Mamede, and H. S. Pinto. Cooking an ontology. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 213–221. Springer, 2006.

A. Rostami, Z. Xie, A. Ishino, Y. Yamakata, K. Aizawa, and R. Jain. World food atlas project. In *Proceedings of the 13th International Workshop on Multimedia for Cooking and Eating Activities*, pages 33–36, 2021.

P. Saengsupawat, T. Angskun, and J. Angkun. Ontology-based knowledge acquisition for thai ingredient substitution. *ARPN Journal of Engineering and Applied Sciences (ARPN)*, 9(10):1461–1468, 2014.

P. Saengsupawat, P. Kitwatthanathawon, T. Angskun, and J. Angskun. Building the thai food ontology for ingredient substitution. In *2017 32th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, 2017.

E. Saenz-Gardea. Identifying sentences with recipe information with natural language processing. In *Kansas State University Undergraduate Research Conference*, 2019.

M. S. Sakib, D. Paulius, and Y. Sun. Approximate task tree retrieval in a knowledge network for robotic cooking. *IEEE Robotics and Automation Letters*, 7(4): 11492–11499, 2022.

A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba. Learning cross-modal embeddings for cooking recipes and food images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3020–3028, 2017.

M. Sam, A. Krisnadhi, C. Wang, J. C. Gallagher, and P. Hitzler. An ontology design pattern for cooking recipes-classroom created. In *WOP*, pages 49–60. Citeseer, 2014.

H. N. Schifferstein, B. M. Kudrowitz, and C. Breuer. Food perception and aesthetics-linking sensory science to culinary practice. *Journal of Culinary Science & Technology*, 20(4):293–335, 2022.

X. Schmitt, S. Kubler, J. Robert, M. Papadakis, and Y. LeTraon. A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 338–343. IEEE, 2019.

O. Seneviratne, J. Harris, C.-H. Chen, and D. L. McGuinness. Personal health knowledge graph for clinically relevant diet recommendations. *arXiv preprint arXiv:2110.10131*, 2021.

N. Shao, P. Murali, and A. Sheopuri. New developments in culinary computational creativity. In *ICCC*, pages 324–327. Citeseer, 2014.

T. Sharma, U. Upadhyay, J. Kalra, S. Arora, S. Ahmad, B. Aggarwal, and G. Bagler. Hierarchical clustering of world cuisines. In *2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*, pages 98–104. IEEE, 2020.

S. S. Shirai, O. Seneviratne, M. E. Gordon, C.-H. Chen, and D. L. McGuinness. Identifying ingredient substitutions using a knowledge graph of food. *Frontiers in Artificial Intelligence*, 3:621766, 2021.

Y. Shirai, W. Kobayashi, K. Takei, K. Yoshizawa, and N. Sato. Discovering new creative mixtures of cooking ingredients. In *Proceedings of the 9th Workshop on Multimedia for Cooking and Eating Activities in conjunction with The 2017 International Joint Conference on Artificial Intelligence*, pages 31–34, 2017.

D. P. Shukla and P. Ailawadi. Computational gastronomy-use of computing methods in culinary creativity. *TRJ Tourism Research Journal*, 3(2):203–211, 2019.

N. Silva, D. Ribeiro, and L. Ferreira. Information extraction from unstructured recipe data. In *Proceedings of the 2019 5th international conference on computer and technology applications*, pages 165–168, 2019.

T. Slimani. Ontology development: A comparing study on tools, languages and formalisms. *Indian Journal of Science and Technology*, 8(24):1–12, 2015.

C. Snae and M. Bruckner. Foods: a food-oriented ontology-driven system. In *2008 2nd IEEE International Conference on Digital Ecosystems and Technologies*, pages 168–176. IEEE, 2008.

C. Spence. Gastrophysics: Getting creative with pairing flavours. *International Journal of Gastronomy and Food Science*, 27:100433, 2022.

H. Su, T.-W. Lin, C.-T. Li, M.-K. Shan, and J. Chang. Automatic recipe cuisine classification by ingredients. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: adjunct publication*, pages 565–570, 2014.

M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López. The neon methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer, 2012.

L. SYSTAP. Bigdata database architecture-blazegraph (2013). *Retrieved Nov*, 29, 2019.

M. M. Taye. Understanding semantic web and ontologies: Theory and applications. *arXiv preprint arXiv:1006.4567*, 22(1):182–192, 2010.

D. Tendall, J. Joerin, B. Kopainsky, P. Edwards, A. Shreck, Q. B. Le, P. Krütli, M. Grant, and J. Six. Food system resilience: defining the concept. *Global Food Security*, 6:17–23, 2015.

R. Tuwani, N. Sahoo, N. Singh, and G. Bagler. Computational models for the evolution of world cuisines. In *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, pages 85–90. IEEE, 2019.

L. R. Varshney, F. Pinel, K. R. Varshney, A. Schörgendorfer, and Y.-M. Chee. Cognition as a part of computational creativity. In *2013 IEEE 12th international conference on cognitive informatics and cognitive computing*, pages 36–43. IEEE, 2013.

L. R. Varshney, J. Wang, and K. R. Varshney. Associative algorithms for computational creativity. *The Journal of Creative Behavior*, 50(3):211–223, 2016.

L. R. Varshney, F. Pinel, K. R. Varshney, D. Bhattacharjya, A. Schörgendorfer, and Y.-M. Chee. A big data approach to computational creativity: The curious case of chef watson. *IBM Journal of Research and Development*, 63(1):7–1, 2019.

R. Venkataramanan, K. Roy, K. Raj, R. Prasad, Y. Zi, V. Narayanan, and A. Sheth. Cook-gen: Robust generative modeling of cooking actions from recipes. *arXiv preprint arXiv:2306.01805*, 2023.

B. C. Villazón-Terrazas, M. Suárez-Figueroa, and A. Gómez-Pérez. A pattern-based method for re-engineering non-ontological resources into ontologies. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 6 (4):27–63, 2010.

D. Vrandečić. Ontology evaluation. In *Handbook on ontologies*, pages 293–313. Springer, 2009.

D. Wang, H. Li, W. Wang, and L. Qiao. Cross-modal knowledge graph construction for multiple food additives. In *Chinese Intelligent Systems Conference*, pages 839–847. Springer, 2022.

J. Youn, T. Naravane, and I. Tagkopoulos. Using word embeddings to learn a better food ontology. *Frontiers in artificial intelligence*, 3, 2020.

Q. Zhang, D. Elsweiler, and C. Trattner. Understanding and predicting cross-cultural food preferences with online recipe images. *Information Processing & Management*, 60(5):103443, 2023. ISSN 0306-4573. doi: https://doi.org/10.1016/j.ipm.2023.103443.