Increasing Diversity through Dynamic Critique in Conversational Recipe Recommendations

Fakhri Abbas
fabbas1@uncc.edu
University of North Carolina at Charlotte
Charlotte, NC, USA

Nadia Najjar
nanajjar@uncc.edu
University of North Carolina at Charlotte
Charlotte, NC, USA

David Wilson
david.Wilson@uncc.edu
University of North Carolina at Charlotte
Charlotte, NC, USA

ABSTRACT

Conversational recommender systems help to guide users to discover items of interest while exploring the search space. During the exploration process, the user provides feedback on recommended items to refine subsequent recommendations. On one hand, critiquing as a way of feedback has proven effective for conversational interactions. On the other hand, diversifying the recommended items during exploration can help increase user understanding of the search space, which critiquing alone may not achieve. Both aspects are important elements for recommender applications in the food domain. Conversational exploration can help to introduce new food items, and diversity in diet has been shown to predict nutritional health. This paper introduces a novel approach that combines critique and diversity to support conversational recommendation in the recipe domain. Our initial evaluation in comparison to a baseline similarity-based recommender shows that the proposed approach increases diversity during the exploration process.

CCS CONCEPTS

• **Information systems** → *Users and interactive retrieval*; **Recommender systems**.

KEYWORDS

Conversational Recommender System, Recipe Recommender, Dynamic Critique, Diveristy

ACM Reference Format:
Fakhri Abbas, Nadia Najjar, and David Wilson. 2021. Increasing Diversity through Dynamic Critique in Conversational Recipe Recommendations. In *Proceedings of the 13th International Workshop on Multimedia for Cooking and Eating Activities (CEA '21), August 21–24, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3463947.3469237

1 INTRODUCTION

Increasing numbers of online sites have emerged over the last decade that enable users to upload and share recipes (e.g., food.com, allrecipes.com). With more users sharing more recipes from different backgrounds, geographic locations, and cultures, there is a vast availability in terms of numbers of recipes, as well as in terms of diversity of cuisines. This has resulted in an overwhelming number of recipes for a user to select from, which can make the search for a suitable recipe time consuming. Moreover, there may be many suitable recipes that the user is not able to see or connect with. To address these issues, a number of recommender system approaches have been proposed for finding recipes more easily [14], and [12].

In the previous studies, given the user's interest, there are three primary ways to recommend recipes, (1) by matching the content of the recipe to the user's interest, (2) by matching the user's interaction with other similar users' interaction, or (3) a combination of both methods. Previous approaches to recipe recommendation have shown promising results in terms of accuracy. However, as noted in [12] and [21], an important aspect that has received comparatively little attention is providing users the opportunity to find diverse sets of recipes.

Incorporating diversity in recommender systems provides several advantages. First, it reduces the problem of "filter bubble" effects [21]. Second, in recipe recommender systems, diversity enables the user to explore alternative options that could be healthier which can increases the dietary diversity for individuals [10]. Lastly, it increases user's awareness and knowledge of existing recipes by providing more recipes that could be explored in different cultures, cuisines, or communities [12]. The problem of incorporating diversity mechanisms in recipe recommendation in order to generate diverse sets that also meet user requirements remains an important open research challenge [12].

One primary aspect of the challenge stems from the differences and similarities between recipes, and users' perception of those differences and similarities. For example, a person may like fried chicken but not grilled chicken, or may prefer chicken for lunch but not for dinner. Due to this complexity in food representation and selection, the recommender system should allow the user to shape the direction of recommendations. This in turn, helps to reduce the effect of contextual factors that are hard to capture such as time, cultural background, food knowledge, and current user's needs. To address this aspect, we propose a conversational recommender approach, in which the user can provide iterative feedback on the recommended items to make the recommendation more tailored to their needs. The second primary aspect lies in promoting dietary diversity and the use of exploration to support this.

This paper addresses the problem of incorporating diversity in a critique based conversational recommender system for recipes. Specifically, this work introduces a novel way of dynamically generating critiques, in which the generated critique leads the user to a more diverse set of recommendations.

CEA '21, August 21–24, 2021, Taipei, Taiwan

ACM ISBN 978-1-4503-8532-9/21/08…$15.00
https://doi.org/10.1145/3463947.3469237

2 BACKGROUND

This work brings together three lines of relevant research: diversity in recommender systems, conversational recommender systems, and recipe recommendation.

2.1 Diversity in Recommender Systems

The concept of diversity and its relationship to accuracy have been studied in Information Retrieval [6] and economics [25] before receiving attention in recommender systems. The idea of diversity introduced by Markowitz in Modern Portfolio Theory models investment as a tradeoff between risk and expected returns [25]. Maximizing the expected return results in higher risk, while diversification of stock portfolios reduces the risk. More generally, recommender systems adopted this idea where ranking items by their predicted relevance increases the risk of producing results that do not satisfy the user because the retrieved items tend to be too similar. On the other hand, diversifying the results reduces risk by increasing the probability of introducing items the user will be interested in [6]. Understanding the balance between accuracy and diversity is an important research question in recommender systems, with the emphasis that users are more likely to be satisfied with diversified results even with a small sacrifice in terms of accuracy [23].

In recommender systems research, diversity has been recognized as the opposite of similarity. Commonly, diversity been measured as the average pairwise distance between items [4]. Ziegler et al. [36] defined the intra-list similarity metric using the aggregate pairwise distances rather than the average. The average pairwise distance is defined by equation 1:

$$Diversity(R) = \frac{\sum_{i \epsilon R} \sum_{j \epsilon R/\{i\}} dist(i,j)}{|R|(|R|-1)} \tag{1}$$

where R is the recommended list of items, and $dist(i, j)$ is the distance between item i and item j.

Using the pairwise distances between items to measure diversity has been widely adopted in the recommender systems literature with variations in the distance metric. These differences in distance metrics depend on the item's representation. For example, when items are represented by its content, the distance has been measured using Jaccard similarity, cosine similarity, or taxonomy-based metric. When items are represented by rating vectors, hamming distance, pearson correlation, and cosine similarity have been adopted as a distance measure [19]. The approach proposed in this paper adopts the view of measuring diversity as the aggregate dissimilarity between items. Hence, the average pairwise distance is used to measure the diversity in the recommended list of items.

2.2 Conversational Recommender Systems

Recommender systems are most often considered as a type of one shot interaction, in which the system recommends a set of items and the user navigates through that set to find an item of interest. Conversely, a conversational recommender system (CRS) takes a different approach, providing a richer interaction with the user through iterative feedback and refinement of results. This in return has a positive impact on enabling users to better understand the search space, and reduce the effect of the cold start problem [18]. McCarthy et al. described CRS as a smart sales assistant, in which the recommender system is seen to play the role of a sales assistant who makes good suggestions. The sales assistant listens to the customer feedback and provides better suggestions in the next round. The agent should be able to provide satisfactory results in a reasonable amount of time as they develop a better understanding of the user's need with each iteration. Similarly the recommender system should as well [27].

The form and the quality of the feedback provided by the user in the CRS determines the success of such recommendation style. Smyth and McGinty in [33] compared between several forms of feedback used in CRS, mainly, Value Elicitation, Ratings-Based, Preference-Based, and Critiquing feedback. In this work, we are interested in the critiquing form of feedback. In the critiquing feedback the user provides a directional preference over a feature of recommendation [7]. For example, in a car recommender system the user might ask for a smaller car engine than the currently recommended car. In this case, the smaller car engine is the critique over the car engine size feature. The form of critique also has variations, a natural language dialog, a system-suggested critiquing, and a user-initiated critiquing [7]. While each form has its own advantages and disadvantages, in this work we are focusing on system-suggested critiquing.

In system suggested critiquing form, as its name suggests, the system generates critique over a set of features. Burke et al. were the earliest in proposing FindMe, a critique based CRS. FindMe helps users to find products through a large multi-dimensional information space [5]. The critiquing in FindMe proposed two challenges, first, it's a pre-designed set of critique and fixed within the user interaction session (static critique), second, each critique can only constraint on one feature (unit critiques). To overcome the first challenge, Reilly et al. have shown the standard critique can be extended to cover multiple features in what's called compound critique [29], For the second mentioned challenge, McCarthy et al. have developed a dynamic critique approach [26], in which the system combines the feature depending on the available items in the search space. In this work, we propose a novel approach to generate dynamic critiques in which each recommended item is presented with a unique set of critiques that is different from critique set of other recommended items. Studies have shown the importance of compounding critique and it's effect on diversity and accuracy [26], but due to space limitation and the current stage of the project this paper focuses on generating dynamic unit critique only.

2.2.1 *Diversity in Conversational Recommender Systems.*

Similar to other recommendation approaches, diversity was applied in conversational recommender systems as well. For example, Kelly and Bridge [20] applied a greedy reranking strategy in a conversational recommender system, the authors diversified the results in each iteration cycle after the feedback was received from the user. In another study, McGinty and Smyth [28] incorporated diversity in the conversational recommender system while balancing the tradeoff between diversity and relevance. The authors described a system where at each cycle, the user selects a critique which is used for the next iteration cycle. The selected item carried over the next recommendation cycle and displayed along other recommended items. If the user selects the carried over item again the system assumes that no progress has been made and a more diversified list is recommended on the next cycle. However, if the user selects a different item, then the system assumes positive progress has been made and generates results with less diversity and more relevance for the next cycle.

This paper proposes a new approach that combines dynamic unit critiquing with diversity in CRS. In comparison to previous studies, our research investigates an approach to generate critiques dynamically by selecting critique that maximizes diversity. The recommendations are generated depending on the user profile and preference, while the critiquing options are generated from diversity scores. Diversity scores are computed using already established diversity metric in literature such as average pairwise distances. Therefore, the recommender system will guide the user through the information space to more diverse items by selecting critiques that takes the user to a more diverse list.

2.3 Recipe Recommender Systems

Recipe and food recommendation is an important application domain for recommender systems, that must address contextual challenges including health/nutrition, mealtime, food cost, and user's mood. Previous recommender systems research for recipes has addressed different aspects. For example, Freyne and Berkovsky made recommendations by representing recipes as ingredients and providing a scoring for each recipe depending on user rating [13]. In another study, they investigated a nearest neighbour approach using Pearson correlation on the rating matrix to provide recommendations [12]. Harvey et al. [17] showed that using SVD provided better performance in comparison to the Freyne and Berkovsky approaches. Ge et al. [14] used a matrix factorization approach by leveraging users' rating and tags to find recipes the user may like.

Food recommendation and exploration is highly dependent on contextual features. To address this, researchers have investigated a variety of context-aware approaches to recipe recommendation. Studies in food and recipe exploration have focused on contextual features such as gender [30], time [22], location [8], and food availability [9]. For example, Cheng et al. [8] explored location contextual factors by filtering users along location. In another study, Ahn et. al [2] studied the factor of culinary cultures in food recommender. Even though several studies addressed food recommender from the contextual factors perspective, understanding which contextual factors are most important in the food selection process remains an open question [31]. In this work, we chose to reduce the effect of contextual factors by providing dynamic critique the enables the user to adjust the recommendation depending on their needs.

2.3.1 Diversity in Recipe Recommender Systems. Grace et al. [15] proposed a system (Q-Chef) that encourages dietary diversity by generating and recommending recipes based on models of surprise and novelty of the ingredients that appear in recipes. While Q-Chef focused on identifying new recipes that are surprising to the user and could result in diversifying their diet, the set of recommended recipes itself is not necessarily diverse. In [11], Elsweiler et al. acknowledged the importance of diversity in meal plans as a way to provide health alternatives. They proposed a meal planner algorithm by recommending recipes to individuals. Despite the acknowledgment they have not engineered diversity into the recommendations. However, the importance of diversity in recipe recommendation has several advantages such as: provides meals with varied sources of nutrition for a balanced meal diet [11], increases user awareness of existing recipes [34], and covers a wide variety of options that could reduce the cold start problem [3].

Overall, while introducing diversity into recipe recommenders has been identified as an important problem in many research papers, the topic has received comparatively little attention. Given that CRS enables the user to lead the direction of recommendation, we chose to adopt this type of recommender system which can address some challenges in the recipe recommendation process such as the user's context and mood.

3 METHODOLOGY: DYNAMIC CRITIQUE GENERATION ALGORITHM

In this section we present our approach for critique generation. This work is an extension to abstract in [1]. The approach consists of four main steps: defining search space, finding maximum diversity, moving from diverse cases to critique, and generating critique text. Figure 1, shows the first three steps in generating critique. We will refer to the figure in more detail while explaining each step.

While the approach is designed for recipe recommendation, it is applicable in other domains with appropriate selection of domain representation and critique features. Recipes in recommender systems can be represented using a variety of different features such as user ratings or ingredients. In this work, we focus on content-based representation rather than user ratings. Therefore, recipes are represented as vectors of ingredients. In terms of critique, we based critique on a recipe's nutritional and flavor features, as they are closely coupled with recipe ingredients.

This section describes the steps of our approach in more general terms: "item" refers to a recipe and "critique" refers to nutritional and flavor features of recipes. More specific details will be covered related to the ingredients, nutritional, and flavour features in the experiment section. The critique generation algorithm presented in this section consists of three steps.

Step 1: Defining search space: Diversity among recommended items is not helpful if the items themselves are not of interest to the user. Thus, the first step to generate meaningful critiques is to define a search space to account for the balance between a baseline user preference and diversity in the recommendation set. Defining the search space among the set of items eliminates outliers, and ensures some similarity between items within the scope of user preference. While limiting the search space may reduce full diversity potential when generating critiques, it ensures the balance between diversity and preference. A very narrow search space will result in less diverse recommended items, while a wide search space may increase the potential for diversity in the recommended items. Figure 1-Step 1, shows a dotted red line representing items within user preference.

Step 2: Finding maximum diversity: The second step is to understand the diversity among the items within this space. In the

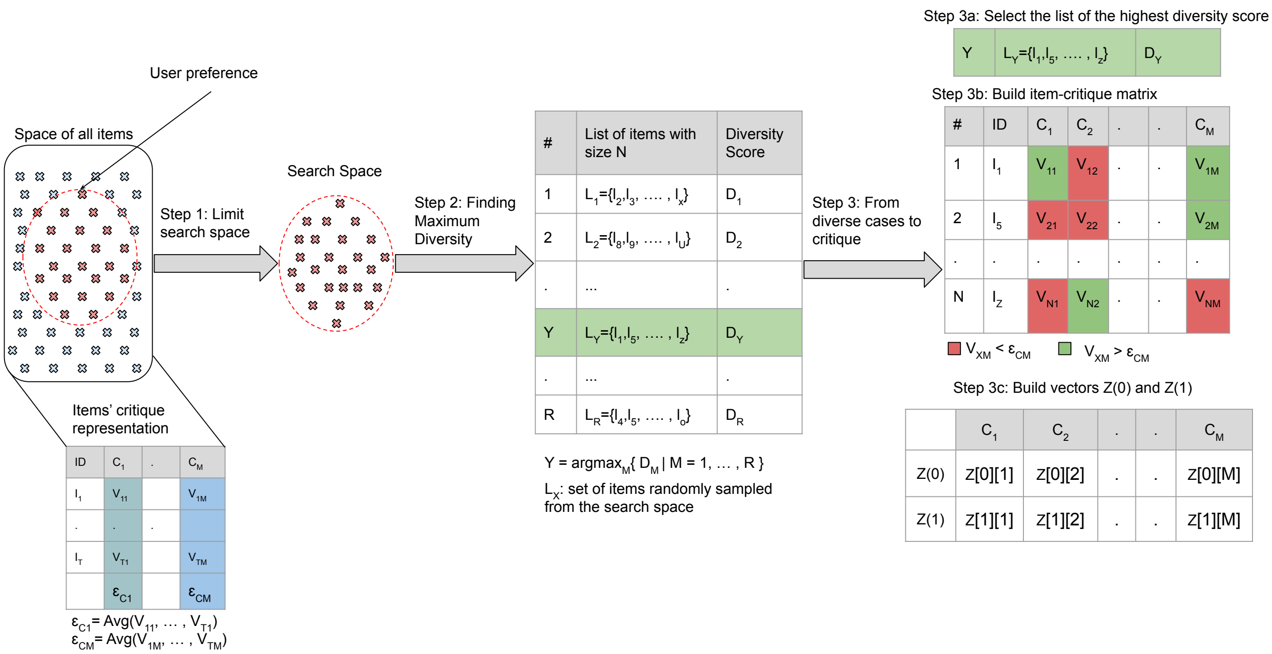

Figure 1: The steps taken to create the Z vectors to generate critique, the text generation step is omitted here.

literature section, we have introduced studies that use pairwise distances to measure the diversity of the list as the aggregate pairwise distance between the items in the list. Similar to previous studies, this work utilizes the average pairwise distances to measure the diversity among the items. Specifically, we randomly select **N** number of items from the search space and measure the average pairwise distances score. This process is repeated **R** number of items. Figure 1-Step 2 shows **R** number of lists with size **N**, each list consist of different items randomly sampled from the search space. Row number **Y** colored in green has the list with the highest diversity score compared to the diversity score of other lists. The resulted table is an insight of the internal composition of items that resulted in high diversity. For example, if the diversity score for list $\mathbf{L_1}$ is higher than the diversity score for list $\mathbf{L_2}$ then this means that, the composition of items in $\mathbf{L_1}$ has more overall diversity compared to $\mathbf{L_2}$. Hence, generating critique by using cases from list $\mathbf{L_1}$ will result in a group of critiques that may increase diversity if one of the critiques is selected in the next recommendation session. This work selects the list of the highest diversity score from the diversity table to be used to generate critique. The higher the value of the parameters **N** and **R** the more likely to find a list $\mathbf{L_x}$ with a high diversity score. However, the value of **N** and **R** are determined empirically, depending on the search space size. The reason behind selecting **N** items randomly, according to Vergas et al [35], is that the maximum diversity can be achieved by selecting items randomly. In addition to that, finding the ideal list of items that generates the maximum diversity comes with a substantial computational cost, therefore a random sampling can achieve the goal with some sacrifice from the optimal solution. The output of this process is a list of items with the highest diversity score.

Step 3: From diverse cases to critique: The third step is to generate potential critique.

Given a set of critiques $\{\mathbf{C_1}, \mathbf{C_2}, \mathbf{C_3}, \dots \mathbf{C_M}\}$, and an item-critique matrix with size of **N**x**M**, where **N** equals the number of items in the list of high diversity score (see previous step), and **M** is the number of critique, a vector **Z(0)** with size **M** can be constructed where each element in the vector holds a percentage of items within the item-critique matrix that has a value *lower* than a threshold value $\mathbf{C_n} \cdot \mathbf{C_n}$ is a pre-computed value for each critique $\mathbf{C_n}$ that can be driven from a statistical measure such as average, or based on domain knowledge. Defining threshold value for each critique feature enables us to transfer the critique column into categorical variable instead of numerical variable. A compliment vector **Z(1)** can be constructed as well, where **Z(1)** = 1 - **Z(0)**. Figure 1 Step 3 shows the three main steps to build the vectors **Z(0)** and **Z(1)**. Step 3a, shows the selected list of items from previous step, $\mathbf{L_M}$ in the figure. In Step 3b, an item-critique matrix is built. In this step, each item represented as a vector of size **M**, where each cell in the vector represents a critique value. For example, $\mathbf{V_{12}}$ is the critique value for item $\mathbf{I_1}$ and critique $\mathbf{C_2}$. The last step in the figure (Step 3c), calculates the percentage of items in the item-critique matrix that has values lower than the pre-defined threshold value for a particular critique, then assign the value to the corresponding column in the vector **Z(0)**. Equation 2 describes how to build the vector **Z(0)**:

$$Z(0)[i] = \frac{count(V_{ij})}{N} \tag{2}$$

such that $V_{ij} < \epsilon_{cj}, 1 \le j \le M$, and $1 \le i \le N$

Creating both vectors to describe the items in the list provides a description of the list $\mathbf{L_M}$ in a consolidated way. In other words, these two vectors provide a percentages for features that when combined together will result in a high diversity score. The features represented in vector **Z** (either **Z(1)** or **Z(0)**) are the source for the possible critique values. For example, a possible critique for an item could be 'less than $\mathbf{C_1}$' where $\mathbf{C_1}$ represents the first feature in vector **Z**. In this work, the critique consists of two components: the first component is the feature, $\mathbf{C_1}$ in our example, and the second component is the direction, 'less than' in this example. Selecting the candidate critique is application dependent since some critique could be more important than other critique, however, we define the set of candidate critiques as the highest **K** vales in each vector (**Z(1)**, and **Z(0)**). Given that both vectors are complementary to each other, high values of **K** will include all features, and small values of **K** will include a proportion of the features. The list of selected features are the pool for candidate critiques.

Step 4: Generate critique text for a given item: To generate critique for item **I** within the search space, the item is represented as a vector of features with length equal to the number of critique (**M**). Each element in the vector **I** is a binary value (0,1), where the element of value 1 means the feature value is above the threshold value $\mathbf{C_n}$ and a value of 0 means the feature value is below the threshold value $\mathbf{C_n}$. Since the purpose of this work is to generate critique that promotes diversity upon selection, we will create an inverse vector of **I**, **I'= 1 - I**. For each element in **I'** if the value is 0 and the feature among the top **K** features in vector **Z(0)** then the feature is selected as a critique. Similarly, if the value is 1 and the feature is among the top **K** features of vector **Z(1)** then the feature is selected as a critique. For the critique direction, critiques that are selected from vector **Z(0)** will be given a direction that indicate 'less', while critiques that are selected from vector **Z(1)** will be given a direction that indicates 'more'.

To illustrate the idea of text generation, Figure 2 shows the process of text generation for two recipes: 'flaky oatmeal raisin cookies' and 'zesty oatmeal raisin cookies'. In this example, the possible critique options are: calories, total fat, sugar, protein, sodium, sat fat, and carbs. The critique values for the recipe 'flaky oatmeal raisin cookies' was transformed into categorical values based on the precomputed threshold values. The threshold values for each critique were simply taken to be the average values for the critique in the search space. For example, the recipe 'flaky oatmeal raisin cookies' has more calories in average compared to all other recipes in the search space, but it has less protein. Since our algorithm is designed to increase diversity we took the inverse of the recipe's vector **I** i.e **I'**. In this example, we selected the top 3 values in vectors **Z(0)** and **Z(1)** (colored in red) as candidate critique (**K=3**). Since the value of 'total fat' critique among the top 3 in the **Z(0)** vector and the inverse of the recipe shows a zero value for the total fat then we select total fat as a critique. For the direction, we chose the text 'less' because the source of the match happened from the vector **Z(0)**. Therefore, the critique text will be 'Less total fat'. The same idea can be used for the next recipe in the Figure 'zesty oatmeal raisin cookies'. In this example, all three features were selected from the vector **Z(1)**. Therefore, the critique text is: More 'sugar', More 'Sat Fat', and More 'Carbs'. Note that, each recipe received

	Calories	Total Fat	Sugar	Protein	Sodium	Sat Fat	Carbs
Z(1)	0.15	0.13	0.27	0.05	0.05	0.18	0.19
Z(0)	0.85	0.87	0.73	0.95	0.95	0.82	0.81
Example 1: Flaky oatmeal raisin cookies recipes							
I: flaky oatmeal raisin cookies	1	1	1	0	0	1	1
I' (Inverse of I)	0	0	0	1	1	0	0
Critique Selection		✖					
Critique Text	Less 'Total Fat'						
Example 2: Zesty oatmeal raisin cookies							
I: zesty oatmeal raisin cookies	0	0	0	0	0	0	0
I' (Inverse of I)	1	1	1	1	1	1	1
Critique Selection			✖			✖	✖
Critique Text	More 'Sugar', More 'Sat Fat', More 'Carbs'						

Figure 2: Two examples of generating critique for the recipe 'Flaky oatmeal raisin cookies' and the recipe 'Zesty oatmeal raisin cookies'

a different critique text using the same vectors **Z(0)** and **Z(1)** that are generated from the previous steps.

4 EVALUATION

As an initial evaluation of our approach, we conduced a simulation study for critique generation. The purpose of the simulation study is (1) to understand the diversity scores of the recommended recipes over the course of the users' interaction, and (2) the feasibility of the proposed algorithm. The following sections describe the recipe dataset used, the experimental setup, and the evaluation results.

4.1 Recipe Dataset

In this work, we chose a recipe dataset that has a potential of diversity. In [32], Sajadmanesh et al. prepared a dataset with 120K recipes crawled from yummly.com, a personalized recipe recommender platform. The dataset consists of recipes from 204 countries. Each recipe has average review rating, ingredients, preparation time, course type, nutritional values, and flavor features. The raw data contains 11,113 ingredients. The course type feature has values related to the recipe type such as afternoon tea, bread, breakfast etc. The nutritional values features are saturated fat, trans fat, fat, carbohydrate, sugar, calories, fiber, cholesterol, sodium, and protein of a recipe per serving. Recipes are identified by six flavors, namely, saltiness, sourness, sweetness, bitterness, spiciness, and savoriness. The flavour features are represented on a scale from 0 to 1.

Given the close coupling between ingredients, flavor, and nutritional values, for this study we use the ingredients to represent the recipes directly to calculate the diversity scores, while the flavor and nutritional features are used as critique features between recipes. To reduce overall sparsity in ingredients we have used the FOODON [16] ontology to map each ingredient to a food concept. The mapping reduced the number of unique ingredients from 11,113 ingredients to 3,807 ingredients. We have grouped recipes by region into 9 regions: Caribbean, North America, South America, Europe,

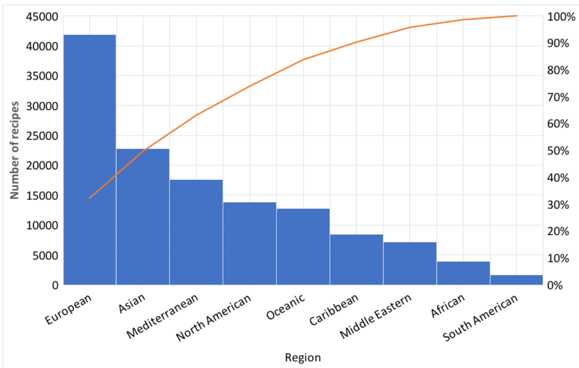

Figure 3: The distribution of recipes over the region in the dataset used for the experiment

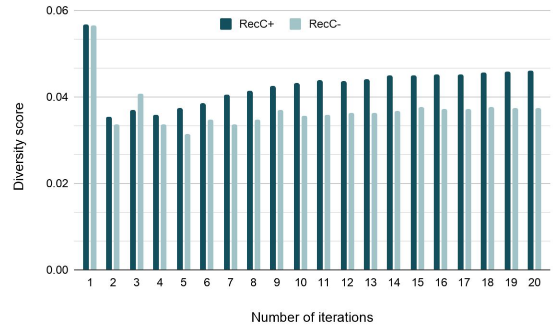

Figure 4: A comparison between RecC+ and RecC- diversity score for the first 20 iterations for the same user

Africa, Middle East, Mediterranean, Asia, and Oceanic. Figure 3 shows the distribution of recipes over these regions.

4.2 Experiment Setup

To evaluate our critique approach, we have built two variations of a recipe recommender system. One variation without critique (RecC-) while the other is with critique (RecC+). The algorithm to recommend recipes in both variations is the same; the only difference is the presence of critique. The first iteration in the recommendation starts recommending the closest N recipes to the centroid of the user search space, where the centroid represents the average score for the ingredients vector. For each iteration, given a selected recipe from the previous iteration the algorithm selects the closest N recipes to the selected recipe. The closest N recipes were determined using the cosine similarity metric, while each recipe is represented by a vector of 3,807 ingredients. In the case of RecC+, we have used the algorithm proposed above to generate critique for each recipe. After the selection of a recipe and a critique, RecC+ recommends N closest recipes to the selected recipe with the critique applied.

To comprehensively evaluate the impact of diversity in recommendation on users of this approach, we recognize that user study evaluation will be needed. As a first step in that direction, in this initial study we have conducted an offline simulation to serve as a baseline for a more comprehensive user study. The simulation consists of building 100 user profiles. Each profile is evaluated by simulating 50 iterations of using RecC+, and RecC-. Since the yummly.com dataset does not provide user interaction with recipes, we have created user profiles using a dataset from the food.com website [24]. The food.com dataset contains user ratings for 180K recipes for 2 million users. We have randomly selected 100 users, where each user had rated more than 50 recipes with 4 or 5 rating (on a typical 1 - 5 scale). Using the FOODON ontology, each recipe found was transformed to a vector of 3,807 features to match the recipes in the yummly.com dataset. Then, a region was assigned to each recipe according to the closest distance between the recipe's vector and the region's centroid vector using cosine similarity. To create a search space for each user, we have built a search space of 25K recipes from the yummly.com dataset that is similar to the user profile. In which, the percentages of recipes per regions in the search space is similar to the percentages found in the user profile. To simulate user selection of recipes at each iteration, we have selected the closest recipe using cosine similarity to the user profile centroid. In the case of RecC+, we have selected a random critique among the critique list for the selected recipe.

For the implementation of RecC+ and RecC-, we used the following settings: $\mathbf{N} = 10$, $\mathbf{K} = 3$, $\mathbf{L} = 100$, $\mathbf{R} = 100$. The total number of unique critique is 16 (6 flavour + 10 nutrition). The threshold value $\mathbf{c_n}$ for each critique were selected to be the average value of the feature represents the critique. To ensure the reproducibility of the results, we have used the user's unique identifier as the random seed in the cases where we used randomness.

4.3 Diversity Improvement Analysis

After running the simulation, at each iteration the diversity score for the recommended set of recipes was measured using the diversity equation 1. Jaccard similarity was used to calculate the pairwise similarity between recipes for each list. For example, Figure 4 shows the diversity score for RecC+ (dark blue), and RecC- (light blue) for the first 20 iterations for one user. As shown in the figure, in all iterations (excepts the first iteration and the third) the diversity score from RecC+ is always higher than the diversity score in RecC-. A Wilcoxon Signed-Ranks Test shows a significant difference in the average diversity score per user between RecC+ and RecC-, $Z=25.0$, $p<0.05$. This indicates that our proposed approach is effective in increasing the diversity for the recommended set over a similarity-based recommender.

4.4 Diversity Improvement and Number of Iterations

The previous analysis shows a statistical significance of the results for a simulation of 50 iterations. In addition, in terms of the conversational interaction, we wanted to investigate the diversity performance over time (conversational iterations). In particular, we wanted to understand after how many iterations the diversity of the recommended recipes in RecC+ would significantly surpass RecC-. Figure 5 shows on the horizontal axis the number of iterations and

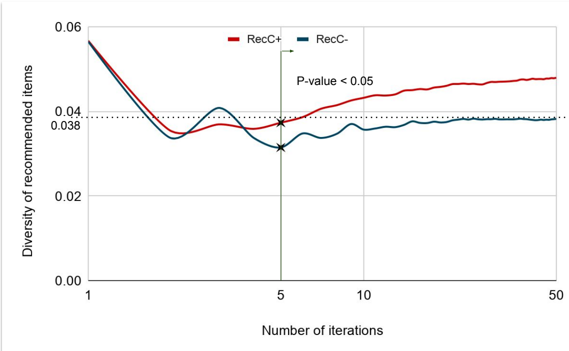

Figure 5: A comparison between RecC+ and RecC- diversity score with the number of iterations for all users

the vertical axis shows the average diversity score of the recommended recipes at the corresponding iteration. The figure shows that at the first three iterations the diversity score for RecC+ and RecC- remained fairly close. But after the fourth iteration RecC+ began to recommended more diverse recipes compared to RecC-, and after the fifth iteration RecC+ is consistently significantly better than RecC- (p-value < 0.05) in terms of diversity. This observation suggests that RecC+ can be used in practical settings, in which after few recommendation cycles the user starts to experience more diverse lists of recipes. The figure also shows that RecC- plateaus after 14 iterations with diversity score no more than 0.038, in the meanwhile the diversity score of the recommended recipes in RecC+ keeps increasing. We note that in both methods a higher diversity score is present at the first iteration, but this is due to the fact that at the first iteration we used the recipes in the user profile for the recommendation, while in the next iterations we used the selected recipe as the seed for the similarity rather than using the user profile.

5 CONCLUSION AND FUTURE WORK

Incorporating diversity in recipe recommenders enables users to explore alternative options that could be healthier which can increases the dietary diversity for individuals. In this work, we have presented a novel approach to generate critiques dynamically in a CRS. The generated critique guides the user through the search space to explore more diverse items. We have implemented the algorithm in a recipe recommender system and conducted an offline analysis to measure the effect on diversity. Our initial results show an improvement in the overall diversity score in comparison to a similarity baseline. This study provides a foundation for further research on diversity critique based recommender approaches in the recipes domain. In future work, we will study the effect of incorporating diversity on recipe recommendation by conducting a controlled user study.

REFERENCES

[1] Fakhri G Abbas, Nadia Najjar, and David Wilson. 2021. Critique Generation to Increase Diversity in Conversational Recipe Recommender System. In *The International FLAIRS Conference Proceedings*, Vol. 34.

[2] Yong-Yeol Ahn, Sebastian E Ahnert, James P Bagrow, and Albert-László Barabási. 2011. Flavor network and the principles of food pairing. *Scientific reports* 1, 1 (2011).

[3] Carl Anderson. 2018. A survey of food recommenders. *arXiv preprint arXiv:1809.02862* (2018).

[4] Keith Bradley and Barry Smyth. 2001. Improving recommendation diversity. In *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland*. Citeseer.

[5] Robin D Burke, Kristian J Hammond, and Benjamin C Young. 1996. Knowledge-based navigation of complex information spaces. In *Proceedings of the national conference on artificial intelligence*, Vol. 462.

[6] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*.

[7] Li Chen and Pearl Pu. 2012. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction* 22, 1-2 (2012).

[8] Hao Cheng, Markus Rokicki, and Eelco Herder. 2017. The influence of city size on dietary choices and food recommendation. In *Proceedings of the 25th conference on user modeling, adaptation and personalization*.

[9] Munmun De Choudhury, Sanket Sharma, and Emre Kiciman. 2016. Characterizing dietary choices, nutrition, and language in food deserts via social media. In *Proceedings of the 19th acm conference on computer-supported cooperative work & social computing*.

[10] Larissa S Drescher, Silke Thiele, and Gert BM Mensink. 2007. A new index to measure healthy food diversity better reflects a healthy diet than traditional measures. *The Journal of nutrition* 137, 3 (2007).

[11] David Elsweiler, Morgan Harvey, Bernd Ludwig, and Alan Said. 2015. Bringing the" healthy" into Food Recommenders.. In *DMRS*.

[12] Jill Freyne and Shlomo Berkovsky. 2010. Intelligent food planning: personalized recipe recommendation. In *Proceedings of the 15th international conference on Intelligent user interfaces*.

[13] Jill Freyne, Shlomo Berkovsky, and Gregory Smith. 2011. Recipe recommendation: accuracy and reasoning. In *International conference on user modeling, adaptation, and personalization*. Springer.

[14] Mouzhi Ge, Mehdi Elahi, Ignacio Fernaández-Tobías, Francesco Ricci, and David Massimo. 2015. Using tags and latent factors in a food recommender system. In *Proceedings of the 5th International Conference on Digital Health 2015*.

[15] Kazjon Grace, M. Maher, D. Wilson, and N. Najjar. 2017. Personalised Specific Curiosity for Computational Design Systems.

[16] Emma J Griffiths, Damion M Dooley, Pier Luigi Buttigieg, Robert Hoehndorf, Fiona SL Brinkman, and William WL Hsiao. 2016. FoodON: A Global Farm-to-Fork Food Ontology.. In *ICBO/BioCreative*.

[17] Morgan Harvey and David Elsweiler. 2015. Automated recommendation of healthy, personalised meal plans. In *Proceedings of the 9th ACM Conference on Recommender Systems*.

[18] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2020. A Survey on Conversational Recommender Systems. *arXiv preprint arXiv:2004.00646* (2020).

[19] Marius Kaminskas and Derek Bridge. 2016. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7, 1 (2016).

[20] John Paul Kelly and Derek Bridge. 2006. Enhancing the diversity of conversational collaborative recommendations: a comparison. *Artificial Intelligence Review* 25, 1-2 (2006).

[21] Bart P Knijnenburg, Saadhika Sivakumar, and Daricia Wilkinson. 2016. Recommender systems for self-actualization. In *Proceedings of the 10th ACM Conference on Recommender Systems*.

[22] Tomasz Kusmierczyk, Christoph Trattner, and Kjetil Nørvåg. 2015. Temporal patterns in online food innovation. In *Proceedings of the 24th international conference on world wide web*.

[23] Michael Madary and Thomas K Metzinger. 2016. Real virtuality: a code of ethical conduct. Recommendations for good scientific practice and the consumers of VR-technology. *Frontiers in Robotics and AI* 3 (2016).

[24] Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating Personalized Recipes from Historical User Preferences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China. https://doi.org/10.18653/v1/D19-1613

[25] Harry Markowitz. 1959. Portfolio selection.

[26] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. 2004. On the dynamic generation of compound critiques in conversational recommender systems. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer.

[27] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. 2005. An Analysis of Critique Diversity in Case-Based Recommendation.. In *FLAIRS Conference*.

[28] Lorraine McGinty and Barry Smyth. 2003. On the role of diversity in conversational recommender systems. In *International Conference on Case-Based Reasoning*.

Springer.
[29] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. 2004. Dynamic critiquing. In *European Conference on Case-Based Reasoning*. Springer.
[30] Markus Rokicki, Eelco Herder, Tomasz Kuśmierczyk, and Christoph Trattner. 2016. Plate and prejudice: Gender differences in online cooking. In *Proceedings of the 2016 conference on user modeling adaptation and personalization*.
[31] Markus Rokicki, Eelco Herder, and Christoph Trattner. 2017. How editorial, temporal and social biases affect online food popularity and appreciation. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 11.
[32] Sina Sajadmanesh, Sina Jafarzadeh, Seyed Ali Ossia, Hamid R Rabiee, Hamed Haddadi, Yelena Mejova, Mirco Musolesi, Emiliano De Cristofaro, and Gianluca Stringhini. 2017. Kissing cuisines: Exploring worldwide culinary habits on the web. In *Proceedings of the 26th international conference on world wide web companion*.
[33] Barry Smyth and Lorraine McGinty. 2003. An analysis of feedback strategies in conversational recommenders. In *the Fourteenth Irish Artificial Intelligence and Cognitive Science Conference (AICS 2003)*. Citeseer.
[34] Youri van Pinxteren, Gijs Geleijnse, and Paul Kamsteeg. 2011. Deriving a recipe similarity measure for recommending healthful meals. In *Proceedings of the 16th international conference on Intelligent user interfaces*.
[35] Saúl Vargas, Linas Baltrunas, Alexandros Karatzoglou, and Pablo Castells. 2014. Coverage, redundancy and size-awareness in genre diversity for recommender systems. In *Proceedings of the 8th ACM Conference on Recommender systems*.
[36] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*.