RecipeMind: Guiding Ingredient Choices from Food Pairing to Recipe Completion using Cascaded Set Transformer

Mogan Gim, Donghee Choi, Kana Maruyama, Jihun Choi, Hajung Kim, Donghyeon Park, Jaewoo Kang

ABSTRACT
We propose a computational approach for recipe ideation, a downstream task that helps users select and gather ingredients for creating dishes. To perform this task, we developed RecipeMind, a food affinity score prediction model that quantifies the suitability of adding an ingredient to a set of other ingredients. We constructed a large-scale dataset containing ingredient co-occurrence based scores to train and evaluate RecipeMind on food affinity score prediction. Deployed in recipe ideation, RecipeMind helps the user expand an initial set of ingredients by suggesting additional ingredients. Experiments and qualitative analysis show RecipeMind’s potential in fulfilling its assistive role in cuisine domain.

CCS CONCEPTS
• Computing methodologies → Knowledge representation and reasoning; • Applied computing;

KEYWORDS
Recipe Ideation, Computational Cooking, Ingredient Set Expansion, Food Affinity Score, Cascaded Set Transformer, Recipe Context

ACM Reference Format:
Mogan Gim, Donghee Choi, Kana Maruyama, Jihun Choi, Hajung Kim, Donghyeon Park, and Jaewoo Kang. 2022. RecipeMind: Guiding Ingredient Choices from Food Pairing to Recipe Completion using Cascaded Set Transformer. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM ’22), October 17–21, 2022, Atlanta, GA, USA. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3511808.3557092

1 INTRODUCTION
Professional chefs and home cooks have pursued to create new dishes and formulate novel recipe ideas which are important tasks in culinary domain. Some recipe ideas derive from existing dishes while others are created from novel ingredient combinations. As a recipe comprises a set of ingredients and sequence of cooking instructions, one may want to brainstorm a recipe starting with a set of ingredients and consecutively expanding it with additional ones. As illustrated in Figure 1 starting with buttermilk, we define these consecutive steps of selecting ingredients (flour, chocolate chips, baking soda) as recipe ideation, gradually leading to recipe completion (Chocolate Chip Cookies).
Recipe ideation is challenging due to vast space of cooking possibilities and complexity for flavor yet important for creative cooking in culinary domain [44]. As multiple ingredients used for cooking a dish form recipe context [38], choosing the right ingredient requires comprehensive understanding in culinary aspects such as aroma or flavor [13]. A systematic approach towards recipe ideation would involve initiating with the most basic recipe idea consisting few ingredients and iteratively updating it with new additional ingredients that goes well with its overall recipe context.

Computational approaches for assisting recipe ideation process have recently become necessary to solve these issues in culinary domain [21]. For instance, Kitchenette predicts food pairing scores [40] and RecipeBowl retrieves the best ingredient given a nearly completed recipe [24] which are deemed as earliest and latest stage of recipe ideation respectively. A more versatile computational approach deployable in any stages of recipe ideation may be desirable.

In this work, RecipeMind performs the ideation task by quantifying the suitability of adding an ingredient to set of other ingredients. Henceforth, we make the following definitions and formulate the objective of recipe ideation task prior to introducing our proposed model RecipeMind.

- **Ingredient Space** \( \mathbb{U} \): A union space containing available ingredients for recipe ideation.
- **Ingredient Set** \( \mathbb{S} \): An finite subset \( \mathbb{S} \in \mathbb{U} \) containing ingredients. In addition, \( \mathbb{S}_n \) is a \( n \)-sized ingredient set where \( |\mathbb{S}_n| = n \) and \( n \geq 1 \).
- **Additional Ingredient** \( i_a \): A single ingredient to be added to current ingredient set where \( i_a \in \mathbb{U} \).
- **Food Affinity Score** \( y \): A score that quantifies the suitability of adding \( i_a \) to \( \mathbb{S}_n \) resulting \( \mathbb{S}_{n+1} = \mathbb{S}_n \cup \{i_a\} \) where \( y \in \mathbb{R} \).

**Problem 1 (Recipe Ideation Task).** We define the objective of Recipe Ideation Task as finite steps of sequential ingredient set expansion. Each step involves expanding the ingredient set \( \mathbb{S}_n \) by adding another ingredient \( i_a \) which results in \( \mathbb{S}_{n+1} = \mathbb{S}_n \cup \{i_a\} \).

To solve the above problem, we introduce the following two tasks.

- **Task 1 (Food Affinity Score Prediction).** Given an ingredient set \( \mathbb{S}_n \) and additional ingredient \( i_a \), RecipeMind \( f \) predicts the food affinity score between \( \mathbb{S}_n \) and \( \{i_a\} \) through modeling \( y = f(\mathbb{S}_n, i_a) \).

- **Task 2 (Additional Ingredient Recommendation).** Given an ingredient set \( \mathbb{S}_n \), all possible ingredients \( i \in \mathbb{U} - \mathbb{S}_n \) and RecipeMind \( f \), the recommended ingredient \( i_a \) to be added to \( \mathbb{S}_n \) is based on the top-ranked affinity score predictions made by RecipeMind.

\[
i_a = \arg \max_i f(\mathbb{S}_n, i)
\]  

To train our RecipeMind model, we constructed a large-scale dataset where each data instance is defined as \( (\mathbb{S}_n, i_a, y) \). The data instances were built from the ingredient subset co-occurrences in the Recipinator dataset containing 507,834 recipes and 2,391 ingredients deemed as Ingredient Space \( \mathbb{U} \) [27]. The food affinity scores were calculated based on Significant PMI based on Document Count and were applied to our recipe ideation task [15, 16].

We adopted the Set Transformer framework when developing the model architecture of RecipeMind [26]. To help RecipeMind jointly learn cross-relational features between ingredients in \( \mathbb{S}_n \) and \( i_a \), we developed Cascaded Set Transformer using Pooling by Multihead Cross-Attention (PMX).

We evaluated RecipeMind’s food affinity score prediction through baseline and ablation experiments with expanding ingredient subsets including unseen sizes in training set. We further analyzed the recommendation results and attention heatmaps after deploying RecipeMind in example recipe ideation scenarios to explore its understanding in recipe contexts.

As shown in Figure 1, RecipeMind encompasses from food pairing to recipe completion as it chooses the most suitable ingredient choices given any number of ingredients in current set. To the best of our knowledge, this work is the first attempt to introduce a data-driven approach that assists ingredient choices at any stage in recipe ideation. The major contributions of our work can be summarized as follows,

- We formulated a downstream task called recipe ideation which features and food affinity score prediction and additional ingredient recommendation.
- We created a large-scale dataset that contains affinity scores for each pair of \( n \)-sized ingredient set and additional ingredient.
- We developed RecipeMind utilizing Cascaded Set Transformers using Pooling by Multihead Cross-Attention.
- We empirically demonstrated RecipeMind’s robustness in expanding set sizes through experiments and analyzed its understanding in recipe contexts throughout recipe ideation scenarios.

2 RELATED WORKS

2.1 Representation Learning for Recipes

Previous works that have introduced various deep learning approaches for improved representation learning of cooking recipes. Cross-modal or multimodal approaches incorporating recipe texts with images have been introduced where some focused on improving representation learning on recipes [6, 8, 32, 33, 37, 48, 50, 53, 54]. Others utilized these features to improve recipe retrieval tasks [7, 9, 29, 58]. Few works have attempted to apply cross-modal feature learning to recipe numeracy tasks such as predicting calories [28] or food ingredient amounts [18].

Meanwhile, as recipes can be expressed as sets of ingredient, recent works have proposed set representation learning methods to effectively learn recipe-related contextual features [24, 27]. Our work also proposes to apply set representation learning to recipe ideation since it is crucial to understand various recipe context and determine the optimal ingredient to be added for the next step.

2.2 Previous Approaches related to Recipe Ideation

One of the downstream tasks related to our work is food pairing which can be deemed as the fundamental form of recipe ideation. Computational methods for food pairing have been introduced including Kitchenette. These methods utilize statistical co-occurrences

\[\text{The source code and demo web page for RecipeMind are open for public access. (https://github.com/dmis-lab/RecipeMind , https://recipemind.korea.ac.kr)}\]
or chemical similarities of ingredient pairs [1, 39, 40]. Another downstream task related to our work is recipe completion [14, 17, 24]. These methods optimized data-driven models to predict ingredients given partial or nearly completed recipes. Our work encompasses both tasks as recipe ideation aims to guide users’ choices on ingredient additions given different ingredient sets.

Other works have suggested approaches for generating recipes containing cooking instructions and/or ingredients. Various generative models using different modalities of queries such as food images, cooking videos and texts have been developed [19, 22, 43, 52, 57]. Recently, a system for recipe editing considering dietary constraints was introduced [10]. While recipe ideation and generation have common traits such as creativity, RecipeMind guides users’ ingredient choices step by step based on its predicted food affinity scores for all expanding sizes of ingredient sets.

3 DATASET

3.1 Obtaining n-sized Ingredient Subsets from Reciptor Dataset

Table 1 shows the dataset statistics involving n-sized ingredient subsets obtained from the Reciptor dataset. We extracted n-sized ingredient subsets from the Reciptor dataset containing 507,834 recipes which was originally used by Reciptor [27]. We adopted Kitchenette’s approach by removing ingredients whose occurrence count does not exceed 20 [40]. As a result, we obtained 2,391 unique ingredients and used them to generate n-sized subsets based on their co-occurrence statistics in the Reciptor dataset [27].

While the number of possible 2-sized subsets (doublets) in our dataset is \( \binom{2391}{2} = 2,857,245 \), we adopted Kitchenette’s approach by selecting doublets exceeding 5 occurrences in the dataset and obtained 236,297. The same criteria was applied to obtaining 3 and 4-sized subsets which resulted in a total of 1,226,767 among \( \binom{2391}{3} \) and 1,952,345 among \( \binom{2391}{4} \) possible subsets. Furthermore, we obtained all available 5,6 and 7-sized ingredient subsets for only testing purposes. Right side of Table 1 shows the obtained numbers of n-sized ingredient subsets.

3.2 Constructing Data Instances

For each n-sized ingredient subset obtained from the Reciptor dataset, we firstly split each n-sized ingredient set \( S_n \) into a pair of \( S_{n-1} \) and its additional ingredient \( i_d \) where the total number of combinations is \( n \). Therefore, the total number of data instances built from ingredient doublets, triplets and quadruplets is 236,297 \times 2, 1,226,767 \times 3 and 1,952,345 \times 4 respectively. The doublet-based data instances are identical to ingredient pairings as the input ingredient set is a singleton. Left side of Table 1 shows the actual numbers of data instances contained in our dataset.

3.3 Generating sPMIr-based Food Affinity Scores

We generated the affinity scores for each data instances consisting a ingredient set and its additional ingredient. Kitchenette used Normalized Point-wise Mutual Information (NPMI) [5, 40, 49] to formulated food pairing scores. The scores represent the co-occurrence between two different ingredients and intuitively determine whether each ingredient pair is suitable or not.

In this work, we adopted Damani’s Significant PMI based on Document Count (sPMId) score, an improved version of PMI considering statistical significance [15, 16]. The formulation of the sPMId score between words \( x, y \) is mathematically expressed as:

\[
s_{PMId}(x, y) = \log \frac{n(x, y)}{n(x) \cdot n(y)} + \sqrt{\max(n(x), n(y))} \cdot \sqrt{\frac{\ln \delta}{\delta - 1}}
\]

where \( n(x) \) is the number of documents that contain \( x \) at least once, \( N \) is the total number of documents and \( \delta \) is the parameter varying between 0 and 1. Prior to applying the sPMId-based score formulation to our task, we substituted words with ingredient subsets of varying sizes.

We propose a modified approach compatible with ingredient subsets used in the Reciptor dataset [27]. The modifications are the following.

- The documents used to calculate occurrences are the recipes which contain a full list of ingredients used for cooking.
- Given two disjoint ingredient subsets \( X \) and \( Y \) (\( |X \cap Y| = 0 \)), a co-occurrence measure is defined based on their union’s occurrence (\( X \cup Y \)).
- As our task involves adding one ingredient to current to a \( N \)-sized ingredient set, we introduced Significant PMI based on Recipe Count (sPMIr) to formulate affinity scores for training RecipeMind using the above modifications.

The calculation of a sPMIr-based affinity score when adding an ingredient \( i_d \) to \( N \)-sized ingredient set \( S_n \) is expressed as:

\[
s_{PMIr}(S, i_d) = \log \frac{r(S_n \cup \{i_d\})}{r(S_n) \cdot r(\{i_d\})} + \sqrt{\max(r(S_n), r(\{i_d\}))} \cdot \sqrt{\frac{\ln \delta}{\delta - 1}}
\]

where \( r(X) \) is the number of recipes that used ingredient set \( X \) at least once, \( R \) is the total number of recipes, \( \delta \) is set as 0.2 and \( |S_n \cap \{i_d\}| = 0 \) since our task involves adding new ingredients. Unlike the pairing scores originally used in Kitchenette [40], the food affinity scores are not bounded since they are not normalized.
Table 1: Statistics of RecipeMind dataset. The filtered 2391 ingredients were used to obtain $n$-sized ingredient subsets $S_n$ used in the 507,834 recipes from the original Receptr dataset. Only subsets whose ingredient co-occurrence count exceeds 5 were used. The obtained $n$-sized subsets were used to construct data instances for training RecipeMind where each consists of $(n-1)$-sized ingredient set $S_{n-1}$, its additional ingredient $i_a$ and calculated food affinity score $y$. The data instances based on $S_2$, $S_3$, $S_4$ were used for training, validation and testing while the remaining ones based on $S_5$, $S_6$, $S_7$ were only used for testing.

| Ingred. Subset Size | Number of Possible Subsets ($\binom{|S_n|}{2}$) | Number of Obtained Subsets $S_n$ | Number of Data Instances ($|S_{n-1}|, i_a, y$) |
|---------------------|-----------------------------------------------|----------------------------------|-----------------------------------------------|
| $2 (S_2 = S_{n-1} \cup \{i_a\})$ | 2.86e06 | 236,297 | 472,594, 378,074, 23,630, 70,890 |
| $3 (S_3 = S_{n-1} \cup \{i_a\})$ | 2.28e09 | 1,226,767 | 3,680,301, 2,944,239, 184,017, 552,045 |
| $4 (S_4 = S_{n-1} \cup \{i_a\})$ | 1.36e12 | 1,952,345 | 7,809,360, 6,247,504, 390,472, 1,171,644 |
| $5 (S_5 = S_{n-1} \cup \{i_a\})$ | 6.48e14 | 1,567,562 | 7,837,810, 0, 0, 7,837,810 |
| $6 (S_6 = S_{n-1} \cup \{i_a\})$ | 2.57e17 | 897,874 | 5,387,244, 0, 0, 5,387,244 |
| $7 (S_7 = S_{n-1} \cup \{i_a\})$ | 8.76e19 | 439,348 | 3,075,436, 0, 0, 3,075,436 |

Table 2: Examples of data instances from RecipeMind dataset with calculated scores and occurrence counts ([$\#$]) in original dataset. The first two merged rows show ingredient set expansion examples based on calculated food affinity scores (sPMIR) where each example has a list of 3 highest- and lowest-scoring ingredients. For example, given $flour$ and $eggs$, adding $baking soda$ gives the highest score (0.6887) while adding $cucumbers$ gives the lowest (-1.5869). The last merged row shows sPMIR scores being calculated differently based on the selected additional ingredient within ingredient set ($flour$, $eggs$, $baking soda$, $nuts$).

| Ingredient Set Expansion | Ingredient Set $S_n$ [6] | Additional Ingredient $i_a$ [6] | sPMIR for $S_{n+1} = S_n \cup \{i_a\}$ [6] |
|--------------------------|--------------------------|---------------------------------|---------------------------------|
| Adding baking soda to flour, eggs | baking soda [51840] | vanilla [29857] | 0.6887 [7510] |
|                         |                          | nuts [5375]                    | 0.6810 [6941] |
|                         |                          | red wine vinegar [6999]        | -1.2334 [60]  |
|                         |                          | cucumbers [4196]               | -1.5846 [6]  |
| Adding nuts to flour, eggs, baking soda | nuts [5375] | vanilla [29857] | 0.7582 [941] |
|                          |                          | buttermilk [8217]              | 0.7653 [1296] |
|                          |                          | parmesan cheese [29226]        | -1.8916 [5]  |
|                          |                          | garlic powder [21429]          | -1.8991 [6]  |
|                          |                          | garlic cloves [6187]           | -2.1562 [9]  |
| Adding $i_a$ to $S_n$ given a fixed set $S_n$ | baking soda [3150] | baking soda [31540] | 0.5689 [941] |
|                          |                          | eggs [77046]                  | 0.3849 [941] |
|                          |                          | flour [56501]                  | 0.3155 [941] |

3.4 Preliminary Analysis on Generated Food Affinity Scores

Figure 2 shows the distribution of sPMIR scores for each size of $S_{n+1} = S_n \cup \{i_a\}$ where $n \geq 1$. The sPMIR-based food affinity scores calculated based on ingredient doublets ($n=2$) and triplets ($n=3$) show normal distributional behavior. However, the distribution tends to become skewed towards negative as the size increases. The shifting distribution of affinity scores may pose challenges to RecipeMind’s generalization in expanding sets.

Table 2 shows the calculated sPMIR-based affinity scores for example ingredient combinations actually used in the Receptr dataset. Given an ingredient set $flour$ and $eggs$, the top scoring additional ingredients are $baking soda$, $vanilla$ and $nuts$. These three ingredients are known to be popularly used with $flour$ and $eggs$ in baking recipes [41, 47].

Higher co-occurrence counts between $S_n$ and $i_a$ have the tendency to result in higher affinity scores. For example, while $nuts$, $red wine vinegar$ and $cucumbers$ have similar occurrences in the dataset, adding $nuts$ yields the highest affinity score among them due to its co-occurrence with $flour$ and $eggs$. The same applies to adding $vanilla$ compared to $parmesan cheese$, $garlic cloves$ given $flour$, $eggs$ and $baking soda$.

Moreover, adding relatively unpopular ingredients may be compensated with higher scores as long as their co-occurrence with current set of ingredients is relatively higher. For instance, while adding $vanilla$ yields a higher affinity score than $nuts$, an expanded set added with $baking soda$ yields an opposite order of affinity scores. While $nuts$ have been used less than $vanilla$ fourfold, their co-occurrence with $flour$, $eggs$ and $baking soda$ is rewarded by sPMIR-based score formulation.

The pairing scores for $S_2$ are symmetric (i.e. adding $x$ to $y$ and vice versa results the same score). However, different combinations of $(S_n, i_a, y)$, $n > 2$ resulted in different food affinity scores as shown in the last 3 rows of table 2. These characteristics presented the necessity of understanding the inter-relational features between recipe context of $S_n$ and $i_a$.

Inferring the missing relations between $S_n$ and $i_a$ from $S_{n+1} = S_n \cup \{i_a\}$ is important in recipe ideation. We performed ingredient subset-based data split for partitioning the $n$-sized ingredients subsets ($n=2, 3, 4$) into training, validation and test purposes (8:0.5:1.5). The validation data instances were used for searching the optimal hyper-parameters for RecipeMind. We ensured that data instances ($S_{n-1}$, $i_a$, $y$) in different partitions do not share the same ingredient subset $S_n$. The remaining ingredient subsets ($n=5, 6, 7$) were used for only testing purposes.

Moreover, adding relatively unpopular ingredients may be compensated with higher scores as long as their co-occurrence with current set of ingredients is relatively higher. For instance, while adding vanilla yields a higher affinity score than nuts, an expanded set added with baking soda yields an opposite order of affinity scores. While nuts have been used less than vanilla fourfold, their co-occurrence with flour, eggs and baking soda is rewarded by sPMIR-based score formulation.

The pairing scores for $S_2$ are symmetric (i.e. adding x to y and vice versa results the same score). However, different combinations of $(S_n, i_a, y)$, $n > 2$ resulted in different food affinity scores as shown in the last 3 rows of table 2. These characteristics presented the necessity of understanding the inter-relational features between recipe context of $S_n$ and $i_a$.

Inferring the missing relations between $S_n$ and $i_a$ from $S_{n+1} = S_n \cup \{i_a\}$ is important in recipe ideation. We performed ingredient subset-based data split for partitioning the $n$-sized ingredients subsets ($n=2, 3, 4$) into training, validation and test purposes (8:0.5:1.5). The validation data instances were used for searching the optimal hyper-parameters for RecipeMind. We ensured that data instances ($S_{n-1}$, $i_a$, $y$) in different partitions do not share the same ingredient subset $S_n$. The remaining ingredient subsets ($n=5, 6, 7$) were used for only testing purposes.
Figure 3: Model Architecture of RecipeMind. $S_n$, $a_i$ is the set of $n$ ingredients and its additional ingredient $i_a$ both encoded ingredient-wise by a shared MLP. $S_i$ and $a_i$ are intermediate encoded representations while $S_C$ and $a_C$ are final contextual representations used to predict a food affinity score.

4 MODEL ARCHITECTURE OF RECIPEMIND

4.1 Overview

RecipeMind ($f(S_n, a_i) = y$) takes a $n$-sized ingredient set $S_n$ and an additional ingredient $i_a$ as input to predict the food affinity score of the updated $(n+1)$-sized set $S_n \cup \{i_a\}$. It consists of three modules which are the Ingredient Set encoder, Additional Ingredient encoder and Affinity Scoring module. Figure 3 shows an overall description for RecipeMind’s model architecture.

Prior to introducing the model architecture of RecipeMind, we refer to Kitchenette built on Siamese Neural Networks which predicts food pairing scores [40]. One of the main justifications for this model design choice is the homogeneity of two inputs (ingredients). Meanwhile, Recipitor and RecipeBowl employed Set Transformers effective representation learning of recipes [24, 27] where both models handle ingredient sets.

As the additional ingredient $i_a$ can be deemed as a 1-sized ingredient set, we may merge two model design choices. However, we propose an alternative approach by using Cascaded Set Transformers with Pooling by Multihead Cross-Attention.

4.2 Ingredient Set Encoder Module

The initial word-based representations of ingredients in $n$-sized set $S_n$ are encoded to $S_0$ by a 2-layered element-wise multi-layer perceptron (MLP) shared in both encoder modules. We used 300-dimensional FlavorGraph embeddings previously trained based on chemical relationships between food ingredients and flavor compounds [39]. The dimension sizes for the encoded ingredients are uniformly set to $h = 128$.

The shared MLP that takes both representations $S_{n+1} = S_n \cup \{a_i\}$ ingredient-wise as input is mathematically expressed as,

$$H = \sigma(Dropout(Linear_{shared}(S_{n+1})))$$ (4)

$$S_0 \cup \{a_0\} = \sigma(Dropout(Linear_{shared}(H)))$$ (5)

where $S_0 \in \mathbb{R}^{n \times 128}$ and $a_0 \in \mathbb{R}^{128}$ are the $n$ 128-dimensional encoded ingredient embeddings in the current set and 128-dimensional additional ingredient embedding. The weights and bias in $Linear_{shared}$ are $W_{shared} \in \mathbb{R}^{300 \times 128}$, $b_{shared} \in \mathbb{R}^{128}$ and $W_{shared} \in \mathbb{R}^{128 \times 128}$, $b_{shared} \in \mathbb{R}^{128}$ respectively. $\sigma$ is the Rectified Linear Unit (ReLU) activation function while Dropout is dropout layer with probability of 0.025.

The encoded ingredients in current set are propagated through 3 stacked Set Attention Blocks (SAB) followed by Sum Pooling [26]. Each successive SAB uses self-attention mechanism to encode higher order ingredient-ingredient relations and form recipe context based on the set of ingredients [24].

The Ingredient Set encoder in RecipeMind taking an $n$-sized encoded ingredient set $S_n$ as input is mathematically expressed as

$$S_0 = MLP_{shared}(S_n)$$ (6)

$$S_i = SAB_i(S_{i-1}) (i = 1, 2, 3)$$ (7)

$$S_C = \text{SumPool}(S_3)$$ (8)

$S_i \in \mathbb{R}^{n \times 128}$ is a set of $n$ 128-dimensional ingredient embeddings encoded by the $i$th SAB ($SAB_i$). $S_C \in \mathbb{R}^{128}$ is a 128-dimensional contextualized embedding for the current ingredient set. SumPool is a permutation-invariant sum pooling operator.

Each SAB is defined as a Multihead Attention Block (MAB) using the same elements as query, key and value [26, 51] and applies self-attention to a set of elements. The SABs followed by the Sum Pooling operator of the Recipe Idea Encoding Layer are mathematically expressed as,

$$S_{AB_1}(S_{i-1}) = MAB_i(S_{i-1}, S_{i-1})$$ (9)

$$MAB(X, Y) = \text{LayerNorm}(H + \text{RFF}_1(X))$$ (10)

$$H = \text{LayerNorm}(X + \text{RFF}_2(\text{MultiAttn}(X, Y, Y)))$$ (11)

$S_{AB_i}$ is the $i$th Set Attention Block, $MAB_i$ is the $i$th Multihead Attention Block, LayerNorm is layer-wise normalization [3] and RFF is row-wise feedforward layer consisting three consecutive MLPs without Dropout layer and using ReLU as non-linear activation function. MultiAttn is an attention mechanism module with 8 heads [51] where the attention weights in each head given query vectors $Q \in \mathbb{R}^{n_q \times 128}$, key vectors $K \in \mathbb{R}^{n_k \times 128}$ and value vectors $V \in \mathbb{R}^{n_v \times 128}$ are calculated.
The aggregation of ingredient-wise representations $S_3$ from the last SAB is mathematically expressed as follows,

$$S_C = \text{SumPool}(S_3)$$  \hspace{1cm} (12)

where $\text{SumPool}$ is a permutation-invariant Sum Pooling operator that performs element-wise summation of the ingredient representations $i_j \in S_3$.

### 4.3 Additional Ingredient Encoder Module

The authors of Set Transformer framework introduced Pooling by Multihead Attention (PMA) as illustrated in Figure 4b [26]. In this block, element-wise representations in set are aggregated by attending them to a single trainable seed vector. Adopting the methods proposed in PercieverIO and PICASO [23, 56], we devised Pooling learning on $S_3$ through the shared MLP and $a_C$.

We denote this as $\text{Cascaded Set Transformer}$ where $\text{SumPool}$ is a permutation-invariant Sum Pooling operator

$$\text{SumPool}(S_3) = \sum_{j=0}^{[S_3]} i_j$$  \hspace{1cm} (13)

The additional ingredient is refined through the shared MLP and 3 successive PMX blocks where each intermediate representation $a_I$ from the $I$th PMX block and its corresponding set from the $I$th SAB in Ingredient Set encoder is fed to the next $I + 1$ th PMX layer. We denote this as $\text{Cascaded Set Transformer}$ since Ingredient Set and Additional Ingredient encoder are jointly connected.

The Additional Ingredient Encoder module taking the additional ingredient $i_a$ as input is mathematically expressed as,

$$a_0 = \text{MLP}_{\text{shared}}(\{i_a\})$$  \hspace{1cm} (14)

$$a_I = \text{PMX}_I(a_{I-1}, S_{I-1})(I = 1, 2, 3)$$  \hspace{1cm} (15)

$$a_C = a_3$$  \hspace{1cm} (16)

where $a_I \in \mathbb{R}^{128}$ is a 128-dimensional ingredient embedding encoded by the $I$th PMX block ($\text{PMX}_I$) based on cross-attention between $a_{I-1}$ and $S_{I-1}$, and $a_C \in \mathbb{R}^{128}$ is a 128-dimensional contextualized embedding for the additional ingredient.

The $I$th PMX$_I$ block that calculates cross-attention between $S_{I-1}$ and $a_{I-1}$, and outputs the refined representation $a_I$ is mathematically expressed as,

$$\text{PMX}_I(a_{I-1}, S_{I-1}) = \text{MAB}_I(a_{I-1}, S_{I-1})$$  \hspace{1cm} (17)

where the $\text{MAB}_I$ is $I$th Multihead Attention Block that computes the attention weights using $a_{I-1}$ as query vector and $S_{I-1}$ as key and value vectors.

### 4.4 Affinity Scoring Module

The Affinity Scoring module concatenates the final contextual representations vector-wise from both sides of encoding layers ($S_C, a_C$) to predict the ideation score $\hat{y}$ of adding an ingredient to current recipe idea and is mathematically expressed as follows,

$$\hat{y} = \text{MLP}_{\text{score}}(S_C \oplus a_C)$$  \hspace{1cm} (18)

where $\oplus$ is vector-wise concatenation between two contextualized embeddings. $\text{MLP}_{\text{score}}$ is a 2-layered MLP for predicting the food affinity score where only the intermediate layer uses Dropout and ReLU as non-linear activation. The weights, bias in the first and second linear layer of $\text{MLP}_{\text{score}}$ are $W_{\text{score}1} \in \mathbb{R}^{256 \times 128}$, $b_{\text{score}1} \in \mathbb{R}^{128}$ and $W_{\text{score}2} \in \mathbb{R}^{128 \times 1}$, $b_{\text{score}2} \in \mathbb{R}^{1}$ respectively.

### 4.5 Model Training

We used root mean squared error as loss objective for training RecipeMind on predicting the food affinity score $y$ for given $S_n$ and $a_I$. The optimizer used in the training process is Adam [25] where the learning rate and weight decay is set to 1e-4, 1e-5 respectively. All RecipeMind and its ablations were trained to a maximum of 30 epochs with batch size of 1024 and early stopping.

### 5 EXPERIMENTS AND ANALYSIS

#### 5.1 Model Baseline and Ablations

We conducted experiments to evaluate and compare our proposed RecipeMind’s performance on affinity score prediction with other baselines and ablations. We firstly made naive predictions based on the statistic mean or median of all affinity scores in training set.

Since food affinity score prediction is a newly formulated task, we borrowed the architecture used in Kitchenette, Reciptor and RecipeBowl [24, 27, 40]. As Kitchenette can only predict affinity scores when adding an ingredient to a 1-sized ingredient set ($|S_1| \cup \{|i_a\}| = 2$), we only trained and tested it on data instances only based on doublets. In context of RecipeMind, both the Ingredient Set and Additional Encoder Module have weight-sharing MLPs while the Ideation Scoring Module has Wide-and-Deep layer [11].

For comparison involving larger ingredient sets, we replaced the weight-sharing MLPs with the weight-sharing Set Transformers used in Reciptor and RecipeBowl. Unlike Kitchenette, the Affinity...
Scoring Module has the same structure as RecipeMind’s. These baseline models along with RecipeMind were trained and tested on data instances based on 2,3,4-sized subsets (|S ∪ {i_d}| = 2, 3, 4). Furthermore, we performed zero-shot testing on unseen ingredient set sizes (|S ∪ {i_d}| = 5, 6, 7).

The models for our baseline experiment are summarized as the following:

- **Naive guessing by statistics (mean, median)**

- **Kitchenette**

- **Reciptor**: Previously used for recipe representation learning, this set transformer consists 2 Induced Set Attention Blocks (ISABs) using 4 attention heads in their MABs followed by Pooling by Multihead Attention using 2 seed vectors [27]. We denote this as Reciptor for brevity.

- **RecipeBowl**: Previously used for recipe completion task, this set transformer consists 1 ISAB using 2 attention heads followed by Pooling by Multihead Attention using 1 seed vector [24]. We denote this as RecipeBowl for brevity.

We also conducted ablation experiments on RecipeMind’s Encoder module and Set Pooling method in Ingredient Set Encoder module. The model ablations for RecipeMind’s Encoder Modules are the following:

- **PMX, + weight-sharing SABs**: We performed this experiment to assess the benefits of using larger weight-sharing Set Transformers. Note that RecipeMind’s Set Transformer is larger than Reciptor’s or Recipebowl’s. We denote this as RecipeMind w/o Cascaded PMX for brevity.

- **PMX, - SAB, + Rep the Set**: We replaced the Set Transformer related blocks with Rep the Set [45]. Since the original implementation of Rep the Set incurs heavily computational costs, we used an approximated version of it instead. We denote this as Rep the Set for brevity.

- **PMX, - SAB**: We performed this experiment to verify employing deeper permutation-invariant models. This ablated model aligns with Deep Sets [55] as it only uses MLP for encoding elements. We denote this as Deep Sets for brevity.
Step Ingredient Set | Top 3 Recommendations by RecipeMind
--- | ---
1 | carrots, onions
2 | celery, potatoes, bay leaves, cabbage
3 | potatoes, cabbage, beef stew meat, bouillon
4 | cabbage heads, tzatziki, canning salt
5 | cabbage heads, bouillon, tzatziki, stout beer
6 | bouillon, stew beer, chicken tenderloins
7 | seitan, chicken tenderloins, string beans, tzatziki
8 | chicken tenderloins, tzatziki, string beans, ditalini

Figure 7: Ingredient recommendations yielded by RecipeMind starting with carrots and onions. Top 3 recommendations are based on their predicted food affinity scores when being added to current ingredient set. Red colors indicate the initial set of ingredients given to RecipeMind and first step. The final set of ingredients implicate RecipeMind has formed recipe context related to soups and stews.

Step Ingredient Set | Top 3 Recommendations by RecipeMind
--- | ---
1 | buttermilk, flour
2 | baking soda, baking powder, vanilla
3 | eggs, vanilla, shortening
4 | vanilla, cocoa, shortening, sour milk
5 | sour milk, Crisco, vanilla flavoring
6 | chocolate, apple jelly, cherry juice
7 | baking chocolate, apple jelly
8 | cherry juice, candied fruit

Figure 8: Ingredient recommendations yielded by RecipeMind starting with buttermilk and flour. The final set of ingredients implicate RecipeMind has formed recipe context related to bakery products such as cakes.

5.2 Experimental Results on Food Affinity Score Prediction

All experiments were conducted on the same training-validation-test split on data instances. We trained and tested the model with different random seeds 5 times and calculated the mean and standard deviation of evaluation metrics which are root mean squared error (RMSE) and Pearson’s correlation (PCORR).

Figure 5 shows the evaluation results of baseline experiments. As shown in Figure 5 RecipeMind outperformed in almost all expanding sizes of ingredient sets in both RMSE and PCORR while Kitchenette achieved the best RMSE among other models given $|S_{n-1} \cup \{i_d\}| = 2$. While Reciptor and RecipeBowl achieved results for $|S_{n-1} \cup \{i_d\}| = 2, 3, 4$ sub par with RecipeMind, they failed to demonstrate generalization in unseen set sizes $|S_n \cup \{i_d\}| = 5, 6, 7$, falling far behind the naive guessing baselines.

The ablation results shown in Figure 6 further support our design choices for RecipeMind’s Encoder Modules and Set Pooling in its Ingredient Set Encoder module. Comparing with ablations on Set Encoder modules, RecipeMind using Cascaded Set Transformers with PMX showed better results than other Encoder ablations. Comparing with ablations on Set Pooling methods, while PMA or Max Pooling showed similar performance when $|S_{n-1} \cup \{i_d\}| = 2, 3, 4$, both ablations failed to generalize when $|S_{n-1} \cup \{i_d\}| = 5, 6, 7$ which supports our choice of Sum Pooling. Overall, the baseline and ablation evaluative results show RecipeMind’s predictability of food affinity scores in expanding ingredient set sizes including unseen ones ($n = 5, 6, 7$) in the training dataset.

5.3 Qualitative Analysis on RecipeMind

5.3.1 Methods

Throughout recipe ideation, recipe context may change continuously due to different ingredient choices and combinatory features in set of ingredients. To investigate this, we deployed RecipeMind in two recipe ideation scenarios where (carrots, onions) and (buttermilk, flour) are given as initial set of ingredients $S_2$. In each set expansion step of ideation scenario, RecipeMind predicts food affinity scores for all available ingredients $i \in U - S_n$ being added to $S_n$. The highest scoring ingredient is selected and added to current set...
(iₙ = arg max \ f(Sₙ, i), Sₙ₊₁ = Sₙ ∪ \{iₙ\}). We repeated this step 8 times yielding a total of 10 ingredients.

Figures 7 and 8 show the top 3 ingredient recommendations for each step of set expansion in RecipeMind’s ideation scenario. For example in Figure 7 at Step 1, RecipeMind given carrots and onions predicted the highest food affinity score for celery. Subsequently, celery is added to current set which is given to RecipeMind for retrieving another ranked list of ingredient recommendations (potatoes, bay leaves, cabbage) at Step 2.

Figures 9 and 10 show the cross-attention weights represented in heatmaps. The attention weights between \( n \)-sized current ingredient set \( S_n \) and additional ingredient \( i_a \) were extracted from RecipeMind’s last PMX block and averaged head-wise [12, 42]. The \( p \)th row and \( q \)th column represent the \( p \)th ingredient set expansion step and \( q \)th ingredient in current set respectively. For example at Step 1, when top-scoring celery was added to current set of carrots and onions, the cross-attention weights of these two ingredients were calculated as \( 0.567 \) and \( 0.432 \) respectively. At Step 2, when top-scoring potatoes was added, the cross-attention weights were calculated as \( 0.365 \), \( 0.32 \) for carrots and onions respectively and \( 0.315 \) for the previously added celery. The cell colors represent the ingredient’s rank of attention weight (i.e. the higher, the darker).

5.3.2 Case Study 1: Narrowing Recipe Context

In the first ideation scenario shown in Figures 7 and 9, RecipeMind was given an initial set of two vegetables carrots and onions which are used in a variety of dishes such as soups & stews, salads or beef recipes [31, 34, 46]. The attention weights in less skewed distribution from Step 1 (0.567, 0.433) to Step 2 (0.365, 0.320, 0.315) show that RecipeMind hasn’t fully determined one of the 3 recipe contexts.

From Step 3 to Step 5, while the top 3 recommendations are diverse in ingredient categories, RecipeMind predicted higher affinity scores for adding vegetable ingredients. Previously added ingredients potatoes (0.360), cabbage (0.288), cabbage heads (0.372) were consecutively assigned with highest attention weights. We speculated that RecipeMind focuses on common ingredient characteristics in determining food affinities and narrowed its recipe context to either soups & stews or salads.

After being added to the ingredient set at Step 6, bouillon which is the core ingredient of soups has, remained dominantly attentive throughout the rest of ideation (0.287, 0.214, 0.172). Follow-up additions seaweed and chicken tenderloins are possible ingredient alternatives for cooking soups & stews dishes [27]. As RecipeMind has further narrowed down its recipe context to soups & stews, we expect future ingredient additions to be highly relevant to it.

5.3.3 Case Study 2: Main and Supportive Recipe Context

In the first ideation scenario shown in Figures 8 and 10, RecipeMind was given initial set of buttermilk and flour as they are essential ingredients for bakery recipes [4, 35]. From Step 1 to Step 5, RecipeMind suggested more essential ingredients (baking soda, baking powder, eggs, vanilla) while maintaining its main bakery recipe context based on buttermilk.

Throughout ideation, the attention weights assigned to buttermilk did not relatively decay as much as the initial ingredients in first scenario (0.694, 0.542, …, 0.185, 0.137). The top 3 recommendations from Step 6 to Step 8 were mostly supportive ingredients such as cherry juice (beverage) and apple jelly (sweet sauces). Moreover, the previously added ingredients cocoa (0.238), sour milk (0.193), baking chocolate (0.172) were consecutively assigned with highest attention weights, implicating their own supportive recipe context. Merging these two recipe contexts will likely lead to chocolate cake recipes [?] served with cherry juice recommended at Step 8.

5.3.4 Summary

We deployed RecipeMind in two recipe ideation scenarios to examine its ingredient set expansion process and interpret the calculated cross-attention weights. Our analysis demonstrates different ingredient choices in recipe ideation lead to different recipe contexts and completion. The attention weights from each of the two ideation scenarios show different changes in RecipeMind’s understanding of recipe context. However, we speculate that RecipeMind is open to vast possibilities of recipe ideation and its completion depending on users’ choices on ingredients.

6 CONCLUSION AND FUTURE WORK

We devised a computational approach for recipe ideation by proposing two tasks; food affinity score prediction and additional ingredient recommendation. We implemented RecipeMind using the Cascaded Set Transformer to help it jointly learn features between current ingredient set and its additional ingredient. We then trained it on our constructed dataset containing food affinity scores. Experimental results including ablations demonstrate RecipeMind’s robustness in predicting affinity scores for expanding ingredient sets. Qualitative analysis provides insight in how RecipeMind understands recipe context in set of ingredients.

While our definition of recipe ideation is confined to adding ingredients to current set, we may expand this into combining two \( n \), \( m \)-sized ingredient sets, creating different combinations of recipe context and deriving novel recipe ideas in the end. Since the food affinity scores are mainly based on co-occurrence statistics, our next step is improving RecipeMind with other important food-related aspects such as nutrition and flavor chemistry. We plan to add nutrition constraints for addressing health benefits and incorporate prior knowledge on flavor chemistry to enhance RecipeMind’s robustness.

Our work is part of a collaboration with Sony AI and their Gastronomy Flagship Project, and the aim is to deploy RecipeMind in food-related applications in order to interactively help chefs create delicious, healthy and sustainable dishes by uncovering vast new opportunities in ingredient combination. We expect RecipeMind to benefit the cuisine domain and accelerate food industry development in the future.

ACKNOWLEDGEMENTS

This research was supported by the National Research Foundation of Korea (NRF-2020R1A2C3010638), the MSIT(Ministry of Science and ICT), Korea, under the ICT Creative Consilience program(IITP-2021-0-01819) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) and Sony AI (https://ai.sony)
REFERENCES
[1] Yong-Yeol Ahn, Sebastian E Ahnert, ... generation network for recipe generation from images. InEuropean Conference on Computer Vision. Springer, 359–374.
[10] Meng Chen, Xiaoyi Jia, Elizabeth Gorbonos, Chinh T Hoang, Xiaohui Yu, and Peter Coucquyt. 2020. Fire
[14] Paula Fermín Cueto, Meeke Roet, and Agnieszka S
[17] Marlies De Clercq, Michiel Stock, Bernard De Baets, and Willem Waegeman. 2019. Implicit Alignment for Cross-Modal Retrieval on Cooking Recipes and Food Images.
[28] Jiatong Li, Ricardo Guerrero, and Vladimir Pavlovic. 2019. Deep cooking: Pre-
[29] Lin Li, Ming Li, Zichen Zan, Qinge Xie, and Jianqian Liu. 2021. Multi-Subspace Implicit Alignment for Cross-Modal Retrieval on Cooking Recipes and Food Images. Association for Computing Machinery, New York, NY, USA, 3211–3215. https: //doi.org/10.1145/3459637.3462189
[30] [36] [chocoflake Blair Lornaman. [n. d.]. One Bowl Buttermilk Chocolate Cake. https: //www.theseasonedmom.com/one-bowl-buttermilk-chocolate-cake/. Accessed: 2022-05-17.
[37] Deborah Madison. 2013. Vegetable Literacy: Cooking and Gardening with Twelve Families from the Edible Plant Kingdom, with over 300 Deliciously Simple Recipes [A Cookbook]. Ten Speed Press.
[38] Luciano DS Paci
[39] Donghyeon Park, Keonwoo Kim, Seoyoon Kim, Michael Spranger, and Jaewoo Kang. 2021. FlavorGraph: a large-scale food-chemical graph for generating food representations and recommending food pairings. Scientific Reports 11 (2021).
[40] Donghyeon Park, Keonwoo Kim, Yonggyu Kim, Jungwoon Shin, and Jaewoo Kang. 2019. Kitchenetene: Predicting and ranking food ingredient pairings using siamese neural networks. In 28th International Joint Conference on Artificial Intelligence, IJCAI 2019 International Joint Conferences on Artificial Intelligence. 5930–5936.
[41] Ruth Petermann. 2022. Desserts. Technical Report. Oklahoma Cooperative Extension Service.
[42] Alessandro Raganato, Jörg Tiedemann, et al. 2018. An analysis of encoder representations in transformer-based machine translation. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. The Association for Computational Linguistics.
[43] Amaia Salvador, Michal Drozdzal, Xavier Giró-i Nieto, and Adriana Romero. 2019. Inverse cooking: Recipe generation from food images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 10463–10462.
[44] Nik Sharma. 2020. The Flavor Equation: The Science of Great Cooking Explained in 5930–5936.
[45] Konstantinos Skianis, Giannis Nikolentzos, Stratis Limnios, and Michalis Vazirgiannis. 2020. Rep the set: Neural networks for learning set representations. In International Conference on Artificial Intelligence, IJCAI 2019. International Joint Conferences on Artificial Intelligence.
[46] Joanne Smith. 2021. Beef. In The Flavor Equation: The Science of Great Cooking Explained in 5930–5936.
[47] Chun-Yuen Teng, Yu-Ru Lin, and Lada A Adamic. 2012. Recipe recommendation using ingredient networks. In Proceedings of the 4th Annual ACM Web Science Conference. 298–307.
[48] Yu Sugiyama and Keiji Yanai. 2021. Cross-Modal Recipe Embeddings by Dis- entangling Recipe Contents and Dish Styles. In Proceedings of the 29th ACM International Conference on Multimedia. 2501–2509.
[49] Yijun Tian, Chuxu Zhang, Ronald Metoyer, and Nitesh V Chawla. 2021. Recipe representation learning with networks. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 1824–1833.
[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems. 5998–6008.
[51] Hao Wang, Guosheng Lin, Steven CH Hoi, and Chunjian Miao. 2020. Structure-aware generation network for recipe generation from images. In European Conference on Computer Vision. Springer, 359–374.
[53] Zhongwei Xie, Ling Liu, Lin Li, and Luo Zhong. 2021. Learning Joint Embedding with Modality Alignments for Cross-Modal Retrieval of Recipes and Food Images. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2221–2230.

[54] Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. arXiv preprint arXiv:1809.00812 (2018).

[55] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J Smola. 2017. Deep sets. corr abs/1703.06114 arXiv preprint (2017).

[56] Samira Zare and Hien Van Nguyen. 2021. PICASO: Permutation-Invariant Cascaded Attentional Set Operator. arXiv preprint arXiv:2107.08305 (2021).

[57] Mengyang Zhang, Guohui Tian, Ying Zhang, and Peng Duan. 2021. Reinforcement learning for logic recipe generation: Bridging gaps from images to plans. IEEE Transactions on Multimedia (2021).

[58] Bin Zhu, Chong-Wah Ngo, Jingjing Chen, and Yanhin Hao. 2019. R2gan: Cross-modal recipe retrieval with generative adversarial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 11477–11486.