KitcheNette: Predicting and Recommending Food Ingredient Pairings using Siamese Neural Networks

Donghyeon Park, Keonwoo Kim, Yonggyu Park, Jungwoon Shin and Jaewoo Kang
Korea University
{parkdh, akim, yongqyu, jungwoonshin, kangj}@korea.ac.kr

Abstract
As a vast number of ingredients exist in the culinary world, there are countless food ingredient pairings, but only a small number of pairings have been adopted by chefs and studied by food researchers. In this work, we propose KitcheNette which is a model that predicts food ingredient pairing scores and recommends optimal ingredient pairings. KitcheNette employs Siamese neural networks and is trained on our annotated dataset containing 300K scores of pairings generated from numerous ingredients in food recipes. As the results demonstrate, our model not only outperforms other baseline models but also can recommend complementary food pairings and discover novel ingredient pairings.

1 Introduction
Many chefs, gourmets, and food-related researchers have focused on studying food pairing for decades. There are books [Page and Dornenburg, 2008] [Dornenburg and Page, 2009] featuring a number of food pairings recommended based on accumulated experiences of professional chefs and food gourmets in the culinary world. Since food pairings are made based on the experiences of experts, food pairing itself is subjective and difficult to quantify. In the academic field, some food-related researchers [Ahn et al., 2011] [Ahn and Ahnert, 2013] [Garg et al., 2017] [Simas et al., 2017] focused on determining the qualities of complementary food pairings based on analysis of sharing flavor compounds. However, FlavorDB built by [Garg et al., 2017] contains only a limited number of flavor compounds and natural ingredients and a considerable amount of time and effort is required to analyze the flavor compounds of food ingredient.

In this work, we introduce KitchenNette which is a model based on Siamese neural networks [Koch et al., 2015]. As shown in Figure 1, KitchenNette first trains on our annotated dataset containing more than 300k scores of known pairings, which constitute only 5% of the total possible number of pairings in our dataset. These quantified scores indicate whether each food pair is complementary or not. Then our trained model predicts the scores of unknown pairings consisting of food ingredients that have infrequently or never been used in recipes, which constitute the remaining 95% of the total number of pairings. The three unknown pairings in Figure 1 that our model found are known to be culture specific pairings in Nordic, Japanese, and Mexican cuisine, respectively. To train our model, we constructed our own dataset which contains the golden standard scores of 300k food ingredient pairings obtained from 1M human-generated cooking recipes [Salvador et al., 2017] [Marin et al., 2018]. Here, the amounts and personal preferences of ingredients and the preparation process in cooking were not considered. Our model employs Siamese neural networks with wide&deep architecture designed to learn the relationship of a food pairing. We then conducted experiments to compare it with several baseline models and confirmed that our model KitcheNette outperformed all the other models.

To further evaluate our model’s prediction performance, three qualitative analyses were conducted. First, we analyzed some example cases of food pairings to test whether our model successfully predicts the scores of unknown pairings. Second, we compared the ranking results of commonly used food ingredient pairings recommended by KitchenNette with those in FlavorDB [Garg et al., 2017]. Our ranking results are more reliable and consistent with human food-pairing knowledge, compared with the results of FlavorDB. Third, we compared the food pairing recommendations of our model, which were based on predicted scores, with those of cooking experts [Page and Dornenburg, 2008].
Dornenburg and Page, 2009). We found that most of the recommendations of our model were the same as those of the cooking experts, which demonstrates the accuracy of our model. In addition, our model recommended food pairings with ingredients not commonly featured in recipes. To this extent, our work attempts to broaden the underlying concept of food pairing and introduce a data-driven, deep learning based approach for discovering novel ingredient pairs.

The major contributions of our work can be summarized as follows.

- We propose a data-driven task that discovers new complementary food pairings that have the potential of being used for future recipes.
- We create a large scale dataset that contains the golden standard scores of food ingredient pairings.
- We show that KitcheNette, which uses Siamese neural networks and wide&deep learning achieves high performance in predicting food ingredient pairing scores.
- We verify KitchenNette’s effectiveness in recommending complementary food pairings and discovering new food pairings through qualitative analyses.

2 Related Work

2.1 Food Related Research

Researches on Discovering Food Pairings [Ahn et al., 2011] and [Ahn and Ahnert, 2013] introduce a flavor network where the network’s edge is built based on the number of flavor compounds shared by culinary ingredients. The flavor network is comprised of 381 ingredients and 1,021 flavor compounds. FlavorDB [Garg et al., 2017] combines existing food repositories to provide a larger database with the user-interactive page. Food-bridging [Simas et al., 2017] improves the flavor network [Ahn et al., 2011] by adding additional bridges between two ingredients through a chain of pairwise affinities despite the chemical compound similarity of the two ingredients being low. However, they cover only a limited number of flavor compounds and natural ingredients and some well-known food pairings (e.g., red wine and beef) have very few flavor compounds in common. Our work employs a data-driven method to define the scores of food pairings from the human experiences and find the new food pairings in large-scale.

Researches on Recommending Recipes The Recipe Recommendation [Teng et al., 2012] has been proposed to determine whether a food ingredient is essential in a recipe. Recipe Recommendation uses two different recipe networks that can accurately predict recipe ratings. Also, finding a surprising and plausible ingredient combination is not new. [Grace et al., 2016] Grace and Maher, 2016 combines cased-based reasoning and deep learning to generate new recipe designs. Our work is similar in that it recommends a new combination of food ingredients, but KitcheNette proposes and trains on a silver standard pairing scores on a larger scale and is able to suggest novel pairings that have never been tried before.

2.2 Siamese Neural Networks

Siamese neural networks, [Koch et al., 2015] have been employed in various tasks to learn similarities between two different inputs. Also, some variations of Siamese neural networks have been introduced. [Mueller and Thyagarajan, 2016] proposed the Manhattan LSTM model which takes two sentences as input. The Manhattan LSTM model generates vector representations for each sentence and calculates the similarity between the vector representations using the simple function $\exp(-||h1-h2||)$ where $h1$ and $h2$ are the embedding vectors. [Yuan et al., 2018] proposed a customized contrastive loss function that can be divided into a partial loss function for positive pairs and a partial loss function for negative pairs. The loss function widens the distance between two vector representations of negative pairs while narrowing the distance between two vector representations of positive pairs. While these two works (Mueller and Thyagarajan, 2016; Yuan et al., 2018) use Siamese neural networks and take two different inputs of the same type, [Liang et al., 2018]’s proposed Siamese-based model takes one input as the standard by which the other input is evaluated. The input is evaluated based on its similarity to the input used as the standard. Our model is designed to train semantic relationships of a food pair beyond simple distance-based similarity functions.

3 Dataset

3.1 Dataset Description and Preprocessing

In this work, we utilized Recipe1M [Marin et al., 2018], a dataset containing approximately one million recipes and their corresponding images which were collected from multiple popular websites related to cooking. All content in Recipe1M can be divided into two categories: texts and images. The recipe texts of Recipe1M consist of the following two parts: the list of ingredients and the instructions of a recipe. The Im2Recipe [Salvador et al., 2017] used a bi-directional LSTM based ingredient name extraction module that performs logistic regression on each word in all the lists of ingredients in Recipe1M to extract the ingredient names only apart. For instance, “2 tbsp of olive oil” is extracted as olive-oil. From the instructions for a recipe, [Marin et al., 2018] trains all the vocabulary including ingredient names with the word2vec [Mikolov et al., 2013] fashion. Among a total of 30,167 learned vocabulary, we obtained 3,567 unique ingredient names as shown in Table 1.

| Recipe1M | # of Recipes | # of Vocab | # of Ingredient Vocab |
|----------|--------------|------------|-----------------------|
|          | 1,029,720    | 30,167     | 3,567                 |

Table 1: Statistics of Ingredients and Pairings. Known Pairs consist of ingredients whose occurrence counts are greater than 20. In addition, each known pair has a co-occurrence count of at least 5.

| Ingredient Pairing Dataset | Total # of Possible Pairs | # of Known Pairs | # of Unknown Pairs |
|----------------------------|----------------------------|------------------|-------------------|
|                            | 6,359,961                  | 356,451          | 6,003,510         |

A demo version of user-interactive KitcheNette is available at http://kitchenette.korea.ac.kr
3.2 Food Ingredient Pairing Score Generation

Food Ingredient Pairing Score Dataset

As mentioned in the earlier sections, traditional ingredient pairing methods have relied on human expertise, such as a long experience in the culinary industry or chemical details of food. As a result, the amount of data to define food pairing and perform deep-learning is absolutely small. To deal with this problem, we propose a new silver standard dataset of food pairing scores that 1) enables training deep-learning models on a large-scale and 2) defines if a pair of food is complementary or not on a scale between -1 and 1. We assumed that the co-occurrence information of ingredients from a large recipe corpus may provide insight into how ingredients are combined in each recipe. Within the scope of this study, we did not consider amounts of ingredients nor their cooking procedures since our dataset is based on a statistical co-occurrence information.

Normalized Point-wise Mutual Information

We calculated our golden standard food pairing scores based on point-wise mutual information (PMI) as introduced in [Teng et al., 2012]. The PMI score (1) is the probability of two elements co-occurring \( p(x, y) \), which is compared to the probability of each element occurring separately \( p(x) \) \( p(y) \). The custom score is designed to accurately represent good/bad pairings by penalizing highly popular ingredients such as salt or butter with low co-occurrence pairs. On the other hand, a pair that shows high co-occurrence with less popular ingredients will represent a good, meaningful pair (e.g., wasabi&noris).

\[
\text{pmi}(x; y) = \log \frac{p(x, y)}{p(x)p(y)} \tag{1}
\]

where:

\[
p(x, y) = \frac{\# \text{ of recipes where } x \text{ and } y \text{ occur together}}{\# \text{ of recipes}}
\]

\[
p(x) = \frac{\# \text{ of recipes where } x \text{ occurs}}{\# \text{ of recipes}}
\]

\[
p(y) = \frac{\# \text{ of recipes where } y \text{ occurs}}{\# \text{ of recipes}}
\]

In our work, we used the normalized version of PMI (NPMI [Bouma, 2009]) to better train and fit our regression model. Point-wise mutual information can be normalized between -1 and +1 where -1 (in the limit) is given for never occurring together, 0 for independence, and +1 for complete co-occurrence. Thus, the scores between -1 and 1 intuitively determine if the pair is well suited or not.

\[
\text{npmi}(x; y) = \frac{\text{pmi}(x; y)}{h(x, y)} \tag{2}
\]

where:

\[
h(x, y) = -\log p(x, y)
\]

Generating Ingredient Pairing Dataset based on NPMI Scores

Ideally, we would calculate all the food ingredient pairing scores of 6,359,961 possible pairing \( \binom{3,567}{2} \) generated from 3,567 unique ingredients. However, we found that ingredients that rarely occurred in 1 million recipe texts may act as noisy samples. Also, ingredients that rarely co-occur may lower the performance of the model. Therefore, we removed ingredients whose occurrence count does not exceed 20 and ingredient pairings whose co-occurrence count dose not exceed 5 to build our golden standard known pairs in Table 1.

As a result, we obtained a total of 356,451 valid known pairings. All the other pairings were considered as unknown pairings. The final distribution of food ingredient pairing scores follows the approximated normal distribution. We assume that the ingredient pairing scores in the upper 5% (\( \mu + 2\sigma \)) of the distribution are the top scores and scores lower than 0.274 are in the lower distribution. The scores of the five best and worst ingredients to pair with vanilla are contained in our dataset, as shown in Table 2.

| Ingredient1 [Count] | Ingredient2 [Count] | Co-occurrence | Pairing score |
|---------------------|---------------------|---------------|---------------|
| baking_soda [58,931]| 14,657              | 0.376         |
| cocoa [6,520]       | 2,759               | 0.360         |
| powdered_sugar [26,729]| 6,558 | 0.314         |
| nut [9,090]         | 2,865               | 0.312         |
| chocolate_chips [9,172]| 2,821 | 0.307         |
| onion [191,691]     | 12                  | -0.589        |
| soy_sauce [40,518]  | 6                   | -0.483        |
| salt_and_pepper [46,534]| 14 | -0.479        |
| garlic [46,534]     | 9                   | -0.477        |
| pepper [68,984]     | 26                  | -0.462        |

Table 2: Ingredient Pairing Dataset. The five best
and worst ingredients in our dataset for pairing with vanilla. The The Flavor Bible [Page and
Dornenburg, 2008] recommended pairing chocolate, coffee, cream, ice cream, or sugar with vanilla.

4 Model

4.1 Learning Ingredient Representations

We propose a model that predicts the scores of ingredient pairings. Our model architecture consists of two major components, as shown in Figure 2. The first is the ‘Ingredient Representation Component’ which uses Siamese neural networks [Koch et al., 2015] where two identical multi-layer perceptrons (MLPs) with the same weights each receive a different 300-dimensional word vector representation. Each MLP has two fully connected layers which process the input ingredient vector. Let \( (X_a, X_b) \) be a pair of ingredients represented as 300-dimensional word vectors. \( W_a \) and \( b_a \) are the shared weights and bias of an MLP, respectively, and \( f(\cdot) \) denotes the activation function for non-linearity. We use \( f(\cdot) = \max(\cdot, 0) \) as rectified linear units (ReLUs). The learned representations \( (h_a, h_b) \) of this pair are mathematically expressed as follows:

\[
h_a = f(W_2f(W_1X_a + b_1) + b_2)
\]

\[
h_b = f(W_2f(W_1X_b + b_1) + b_2)
\]

where \( W_1 \in \mathbb{R}^{300 \times 300} \), \( W_2 \in \mathbb{R}^{3 \times 300} \), \( b_1 \in \mathbb{R}^3 \), and \( b_2 \in \mathbb{R}^3 \), and \( i \) and \( j \) are the number of hidden units.

4.2 Predicting Food Ingredient Pairing Scores

In the ‘Pairing Score Prediction Component’, we employ wide&deep learning [Cheng et al., 2016]. The layer is divided into a wide layer and a deep layer. In the deep layer, two \( j \)-dimensional learned representation vectors are concatenated and passed to another MLP that computes a joint representation of two ingredients. This joint representation is denoted as deep vector \( d \) and is mathematically expressed as follows:
We train our proposed model to minimize the loss function which is mathematically expressed as follows:

\[ L(\Theta) = \frac{1}{N} \sum_{a,b} (y_{ab} - Y_{ab})^2 \]

where \( L \) is the computed loss function to be minimized during training, \( \Theta \) are the model parameters to be trained, \( y_{ab} \) is the true score value, \( Y_{ab} \) is the predicted score value, and \( N \) is the total number of input pairs used for training. We use the Adam optimizer for our model.

4.3 Model Training Details

We performed ablation tests to evaluate each feature of KitcheNette. As illustrated in Table 3, the following five metrics were utilized to evaluate model performance: Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Error (MAE), Correlation (CORR), and R squared (R2). Our KitcheNette model clearly outperforms the baseline models in all metrics.

5 Experiment

5.1 Baseline Models

We first evaluated the baseline models before evaluating our proposed model. We first predicted the pairing scores by simply calculating the cosine similarity between two input ingredient vectors. We employed the following machine learning models from the Python Scikit-learn [Pedregosa et al., 2011] package as our baseline models: Linear Support Vector Regressor, Random Forest Regressor, Extra Tree Regressor, SGD Regressor, and Gradient Boosting. Additionally, a simple version of Siamese Neural Network [Koch et al., 2015] is used one of the baseline models. All these models are fitted with hyperparameters estimated by the built-in grid search.

5.2 Main Results

As illustrated in Table 3, the following five metrics were utilized to evaluate model performance: Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Error (MAE), Correlation (CORR), and R squared (R2). Our KitcheNette model clearly outperforms the baseline models in all metrics.

We use Normalized Discounted Cumulative Gain (NDCG@K) to evaluate the ranking performance of our model (Figure 3a) and employ ROC curve to evaluate the sensitivity of our model (Figure 3b). In terms of NDCG@K, our model outperforms all the baseline models in making accurate predictions. The ROC curve is also used to measure the classification performance of the models. As mentioned in Section 3.2, we regarded all pairings with prediction scores of 0.274 or higher as complementary pairings; pairings with lower scores were considered non-complementary. The ROC curve results demonstrate that our KitcheNette model achieves higher performance than all the other models in predicting complementary pairings.

5.3 Ablation Study

We performed ablation tests to evaluate each feature of KitcheNette. As illustrated in Table 4, the wide&deep architecture and ingredient embedding of our model help improve its performance. When our model uses the cosine similarity of learned representations from the Siamese networks, it obtains the lowest performance in predicting food pairing scores. The concatenation (deep layer) of two representations dramatically improves the performance of our model. This indicates that semantic relations need to be learned for predicting food pairing scores. Furthermore, the wide&deep architecture that learns the relation of two ingredients further boosts our model’s performance. Also, utilizing the ingredient embedding for input vectors, instead of randomly initialized vectors, improves the model’s performance.

6 Qualitative Analysis

For qualitative analysis, we considered to perform experiments with actual food and get human feedback, but realized it was not easy to evaluate large-scale pairing scores and beyond scope of our work. Instead, we performed various case studies. On top of that, we provide a demo version of KitcheNette.

\[ d = f(W_4 f(W_3(h_a, h_b) + b_3) + b_4) \]

where \( W_3 \in \mathbb{R}^{j \times 2j} \), \( W_4 \in \mathbb{R}^{j \times 3} \), \( b_3 \in \mathbb{R}^j \), \( b_4 \in \mathbb{R}^j \), and \( j \) is the number of hidden units in each layer. The matrix is then flattened to a \( j^2 \)-dimensional wide vector \( w \) which is mathematically expressed as follows:

\[ w = g(h_a \otimes h_b) \]

where \( g \) denotes a flattening function that converts a \( n \times n \) matrix into a \( n^2 \)-dimensional single vector.

The wide vector \( w \) is then directly concatenated to the deep vector \( d \). The concatenation of the wide and deep vectors is passed to the last fully connected layer to compute the pair score for the final output. Overall, as \( (w, d) \) is the concatenation of the wide and deep vectors, the final output score \( Y \) for the ingredient pair \( (X_a, X_b) \) is mathematically calculated as follows:

\[ Y = W_5(w, d) + b_5 \]

where \( W_5 \in \mathbb{R}^{1 \times (j^2 + j)} \) and \( b_5 \in \mathbb{R}^1 \), and \( j \) is the number of hidden units in each layer.
we retrieved the top 10 ingredient pairings that consist of the selected ingredients. Based on our observations, KitcheNette generally recommended food ingredients that are frequently used in everyday cooking and dining (e.g., tomato&lettuce, onion&ground_beef, pepper&oregano, cin-

where anyone could retrieve the scores of ingredient pairings that they want to find.

6.1 Finding Unknown Pairings

To demonstrate the accuracy of KitcheNette’s predictions on infrequently used food ingredient pairings, we performed a comparative analysis of the prediction results of both known and unknown ingredient pairings. As illustrated in Table 5, we chose three similar carbonated white wines (champagne, sparkling_wine, and prosecco). We then calculated the score of each wine paired with a different ingredient and analyzed all the possible pairings for the three different cases below.

Case 1 We used the champagne&orange_twist as a given pairing for comparison since it is a well-known ingredient pairing with a high annotated score. The orange_twist&sparkling_wine and orange_twist&prosecco pairings do not have annotated scores since they are uncommon pairings. Additionally, we chose two other ingredients (orange_wedge and lime_twist) that are similar to orange_twist, but are not frequently used with any of the three wines. As a result, we have one known pairing and eight unknown pairings. The prediction results of all the nine pairings were consistently high (0.33-0.45).

Case 2 Based on their prediction scores, we paired the wines with different ingredients to create the following three unique known pairings: champagne&elderflower_liqueur, sparkling_wine&cream_de_cassis, and prosecco&lemon_sorbet). The prediction results of the remaining six unknown pairings were also consistently high (0.29-0.42) compared to the three given known pairings.

Case 3 Finally, we chose onion as it made the worst known pairing with champagne and paired it with the remaining two wines. The predictions results of the two unknown pairings (sparkling_wine&onion and prosecco&onion) were consistently as low as the scores of the known pairing (champagne&onion).

In sum, our prediction results show that KitcheNette is capable of making predictions on certain pairings based on analogical reasoning which states that if A is similar to B and A forms a good pairing with C, then it is more likely that B also forms a good pairing with C. We believe that using this reasoning enhanced the performance of KitcheNette on unknown food pairings without annotated scores.

6.2 Comparison of Food Pairing Ranking Results

We performed a comparative analysis between the ingredients ranked by KitcheNette and the ranked ingredients in FlavorDB. We selected four widely used food ingredients (tomato, onion, pepper and cinnamon) to . Then we retrieved the top 10 ingredient pairings that consist of the selected ingredients. Based on our observations, KitcheNette generally recommended food ingredients that are frequently used in everyday cooking and dining (e.g., tomato&lettuce, onion&ground_beef, pepper&oregano, cin-

Table 3: Prediction results of the models.

| Model                | RMSE | MSE | MAE | CORR | R2    |
|----------------------|------|-----|-----|------|-------|
| Cosine Similarity    |      |     |     |      |       |
| Gradient Boosting    | 0.1073 | 0.0115 | 0.0815 | 0.3339 | 0.0773 |
| SGD                  | 0.0993 | 0.0099 | 0.0762 | 0.4585 | 0.2105 |
| Linear SVR           | 0.0993 | 0.0099 | 0.0762 | 0.4588 | 0.2105 |
| Random Forest        | 0.0802 | 0.0064 | 0.0612 | 0.7015 | 0.4846 |
| Extra Tree           | 0.0742 | 0.0055 | 0.0566 | 0.7664 | 0.5586 |
| Siamese Network      | 0.0726 | 0.0054 | 0.0540 | 0.8223 | 0.5679 |
| KitcheNette          | 0.0421 | 0.0018 | 0.0320 | 0.9249 | 0.8551 |

Table 4: Ablation tests on the validation set.

|                      | RMSE | MSE | MAE | CORR | R2    |
|----------------------|------|-----|-----|------|-------|
| KitcheNette          |      |     |     |      |       |
| +Cosine,-Wide&Deep   | 0.0421 | 0.0018 | 0.0320 | 0.9249 | 0.8551 |
| -Wide Layer          | 0.0726 | 0.0054 | 0.0540 | 0.8223 | 0.5679 |
| -Ingredient Embedding| 0.0432 | 0.0019 | 0.0326 | 0.9206 | 0.8474 |
|                      | 0.0501 | 0.0025 | 0.0383 | 0.8923 | 0.7958 |

Table 5: Examples of known and unknown pairings. † and * refer to the predicted scores of known and unknown pairings, respectively.
Onion  tomato  beans  marjoram  dill  
Cinnamon  potato  rosemary  celery  oregano

KitcheNette  FlavorDB

Pepper  bay  hamburger  celery

KitcheNette  FlavorDB

eggplant  apple  ground  canned  applesauce  oregano  garam  ground  brown  nutmeg  nutmeg  green  taco  elbow  cucumber  mango  potato  potato  avocado  potato

KitcheNette  FlavorDB

oregano  ginger  stewing  turmeric  onion  nutmeg  thyme  basil

KitcheNette  Recommendations

We also found that KitcheNette can discover new food-drink pairings, which shows that our data-driven learning model is also able to recommend food ingredients less common in non-Asian cuisines.

### 6.3 Discovering New Drink-Food Pairings

We also found that KitcheNette can discover new food-drink pairings, which we believe is one of the main aims of food pairing. As illustrated in Table 6, we compared our model’s food&drink recommendations with those from “The Flavor Bible” [Page and Dornenburg, 2009] and “WHAT to DRINK with WHAT you EAT” [Dornenburg and Page, 2009]. The recommendations are listed in alphabetical order. *refers to the predicted scores of unknown pairings.

| Rank | KitcheNette Recommendations | Red Wine | White Wine | Gin | Sake |
|------|-----------------------------|----------|------------|-----|------|
| 1    | beef_stock                  | mussel   | butter     | angostura_hitters | apple brandy |
| 2    | beef_cheeks                 | shad*    | chicken    | sweet_vermouth | apricot brandy |
| 3    | lamb_shank                  | cockle   | crab       | benedictine  | basil         |
| 4    | beef_broth                  | shrimp_shells | cream  | orange_bitters  | blackberries |
| 5    | pan_juices*                 | fish_fumeet | fish     | elderflower_liqueur | cerbey  |
| 6    | chicken_rocks               | lobster_base | lobster | orange_twist | champange |
| 7    | salpeter*                   | arborio_rice | salmon | lemon_twist | cilantro |
| 8    | tomato_paste                | skate*   | scallops   | simple_syrup | cola |
| 9    | oxtail                      | oyster_liqueur | shellfish | dry_vermouth | cranberry_juice |
| 10   | dry_red_wine                | cuttlefish* | real     | orange_wedge* | ginger |
| 11   | veal_stock                  | shrimp   |           | aquavit* | herbs |
| 12   | ajinomoto*                  | fish_stock |         | pisco* | lemon juice |
| 13   | pike*                       | mirlitons* |         | eye_whiskey* | lime juice |
| 14   | lamb_stock                  | crayfish* |         | honey_cypur | mint |
| 15   | beef_jones*                 | escargot |         | mezcal* | orange juice |
| 16   | verjusce*                   | scamp*  |         | absinthe | oysters |
| 17   | cherry_cola                 | clam_juice |         | curacao | tonic |
| 18   | beef_stew_seasoning*        | fish_jones |         | campari |         |
| 19   | brisket*                    | lobster_shells* |         | lemon_twists |         |
| 20   | ti_leaves*                  | scallop |           | wheat_starch* |         |

Table 6: The ranking results of KitcheNette and FlavorDB

| Rank | KitcheNette Recommendations | Red Wine | White Wine | Gin | Sake |
|------|-----------------------------|----------|------------|-----|------|
| 1    | beef_stock                  | mussel   | butter     | angostura_hitters | apple brandy |
| 2    | beef_cheeks                 | shad*    | chicken    | sweet_vermouth | apricot brandy |
| 3    | lamb_shank                  | cockle   | crab       | benedictine  | basil         |
| 4    | beef_broth                  | shrimp_shells | cream  | orange_bitters  | blackberries |
| 5    | pan_juices*                 | fish_fumeet | fish     | elderflower_liqueur | cerbey  |
| 6    | chicken_rocks               | lobster_base | lobster | orange_twist | champange |
| 7    | salpeter*                   | arborio_rice | salmon | lemon_twist | cilantro |
| 8    | tomato_paste                | skate*   | scallops   | simple_syrup | cola |
| 9    | oxtail                      | oyster_liqueur | shellfish | dry_vermouth | cranberry_juice |
| 10   | dry_red_wine                | cuttlefish* | real     | orange_wedge* | ginger |
| 11   | veal_stock                  | shrimp   |           | aquavit* | herbs |
| 12   | ajinomoto*                  | fish_stock |         | pisco* | lemon juice |
| 13   | pike*                       | mirlitons* |         | eye_whiskey* | lime juice |
| 14   | lamb_stock                  | crayfish* |         | honey_cypur | mint |
| 15   | beef_jones*                 | escargot |         | mezcal* | orange juice |
| 16   | verjusce*                   | scamp*  |         | absinthe | oysters |
| 17   | cherry_cola                 | clam_juice |         | curacao | tonic |
| 18   | beef_stew_seasoning*        | fish_jones |         | campari |         |
| 19   | brisket*                    | lobster_shells* |         | lemon_twists |         |
| 20   | ti_leaves*                  | scallop |           | wheat_starch* |         |

Table 7: Food&drink Pairings. The ranked pairings of KitcheNette and food&drink recommendations from “The Flavor Bible” [Page and Dornenburg, 2008] and “WHAT to DRINK with WHAT you EAT” [Dornenburg and Page, 2009]. The recommendations are listed in alphabetical order.

For red wine and white wine, our model recommended a variety of meat (e.g., beef, lamb) and specific seafood ingredients (e.g., mussel, lobster, shrimp) respectively. Our model recommended authentic Japanese food ingredients to pair with sake, which shows that our data-driven learning model is also able to recommend food ingredients less common in non-Asian cuisines.

### 7 Conclusion & Future Work

In this work, we introduced KitcheNette which predicts food ingredient pairing scores based on a large amount of food recipe data, and ranks food ingredient pairings based on the predicted scores. Our model which has Siamese deep neural networks is trained on dataset containing more than 300k food ingredient pairing scores. We demonstrate that our model discovers new and unknown pairings and achieves better ranking results than the existing food pairing ranking models. Also, our model discovers new drink-food pairings and accurately predicts the scores of new food ingredient pairings.

For future work, we plan to use a graph-based neural network architecture to train on one-to-many ingredient pairings, instead of on one-to-one pairings, which were used by our model’s Siamese networks. Also, we plan to add the chemical information of food ingredients to the ingredient embeddings and use more detailed information on food ingredients from food encyclopedias. Last, we would like to use more novel and authentic recipes to help our model to recommend more versatile food ingredient pairings.
References

[Ahn and Ahnert, 2013] Yong-Yeol Ahn and Sebastian Ahnert. The flavor network. *Leonardo*, 46(3):272–273, 2013.

[Ahn et al., 2011] Yong-Yeol Ahn, Sebastian E Ahnert, James P Bagrow, and Albert-László Barabási. Flavor network and the principles of food pairing. *Scientific reports*, 1:196, 2011.

[Bouma, 2009] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40, 2009.

[Cheng et al., 2016] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10. ACM, 2016.

[Dornenburg and Page, 2009] Andrew Dornenburg and Karen Page. *What to Drink with What You Eat: The Definitive Guide to Pairing Food with Wine, Beer, Spirits, Coffee, Tea-Even Water-Based on Expert Advice from America’s Best Sommeliers*. Little, Brown, 2009.

[Grace and Maher, 2016] Kazjon Grace and Mary Lou Maher. Surprise-triggered reformulation of design goals. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[Grace et al., 2016] Kazjon Grace, Mary Lou Maher, David C Wilson, and Nadia A Najjar. Combining cbr and deep learning to generate surprising recipe designs. In *International Conference on Case-Based Reasoning*, pages 154–169. Springer, 2016.

[Koch et al., 2015] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[Li et al., 2010] Chun-Yuen Teng, Yu-Ru Lin, and Lada A Adamic. Recipe recommendation using ingredient networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 298–307. ACM, 2012.

[Mikolov et al., 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.