

# Eating Healthier: Exploring Nutrition Information for Healthier Recipe Recommendation

Meng Chen<sup>a</sup>, Xiaoyi Jia<sup>a</sup>, Elizabeth Gorbonos<sup>b</sup>, Chnh T. Hong<sup>b</sup>, Xiaohui Yu<sup>c</sup>,  
Yang Liu<sup>b,\*</sup>

<sup>a</sup>*Shandong Univeristy, 27 Shanda South Road, Jinan, Shandong, China*

<sup>b</sup>*Wilfrid Laurier University, 75 University Avenue West, Waterloo, Ontario, Canada*

<sup>c</sup>*York University, 4700 Keele Street, Toronto, Ontario, Canada*

---

## Abstract

With the booming of personalized recipe sharing networks (e.g., Yummly), a deluge of recipes from different cuisines could be obtained easily. In this paper, we aim to solve a problem which many home-cooks encounter when searching for recipes online. Namely, finding recipes which best fit a handy set of ingredients while at the same time follow healthy eating guidelines. This task is especially difficult since the lions share of online recipes have been shown to be unhealthy. In this paper we propose a novel framework named NutRec, which models the interactions between ingredients and their proportions within recipes for the purpose of offering healthy recommendation. Specifically, NutRec consists of three main components: 1) using an embedding-based ingredient predictor to predict the relevant ingredients with user-defined initial ingredients, 2) predicting the amounts of the relevant ingredients with a multi-layer perceptron-based network, 3) creating a healthy pseudo-recipe with a list of ingredients and their amounts according to the nutritional information and recommending the top similar recipes with the pseudo-recipe. We conduct the experiments on two recipe datasets, including Allrecipes with 36,429 recipes and Yummly with 89,413 recipes, respectively. The empirical results support

---

\*This is to indicate the corresponding authors.

*Email addresses:* mchen@sdu.edu.cn (Meng Chen), jiaxiaoyishd@163.com (Xiaoyi Jia), gorb6620@mylaurier.ca (Elizabeth Gorbonos), choang@wlu.ca (Chnh T. Hong), xhyu@yorku.ca (Xiaohui Yu), yangliu@wlu.ca (Yang Liu)

the framework’s intuition and showcase its ability to retrieve healthier recipes.

*Keywords:* recipe recommender system, healthy recipes, embedding learning, neural network, nutrition

---

## 1. Introduction

The presence of food related content on the web has become prominent in recent years. Along with the overwhelming volume of images on social-media, the use of online sites as a source for recipes and culinary ideas is expanding [1, 2, 3]. With the large quantity of available data, finding the right recipe becomes a difficult task. Food recommender systems, a domain specific subclass of recommender systems, have been designed to assist in providing users with recipes or meal plans that accommodate their preferences. Food recommenders, which usually rely on a tally to indicate people’s aggregated opinion; however are inevitably facing the following challenges that cannot be easily addressed by adopting existing methods. First, the majority of online recipes do not adhere to the nutritional guidelines set by international food agencies and are therefore regarded as unhealthy. Second, a negative correlation is displayed between rating scores and healthiness measures, meaning highly rated recipes tend to be even less healthy [4]. Third, multiple aspects influence our food preferences among which are geographical and cultural affiliation. Being linked to these factors, cuisine recommendation generated based on people’s votes can be a great mean to express group inclinations; however, the level of healthy and individual preference should be captured in other forms.

In contrast, a user’s culinary inclination can be perceived as a result of personal ingredient preferences or some latent factors. Some works use ingredients’ likings, whether in the form of an explicit input [5] or by indirectly learning them from recipe ratings [6]; while others employ standard collaborative filtering approaches to uncover hidden factors [4]. Despite their merits in recommending recipes one would like or suggesting suitable extensions to an ingredient list, they do not consider the *recipe-completion* task (where a complete recipe is

constructed from a list of ingredients) and for the most part they overlook the nutritional aspect.

To meet these challenges, we propose NutRec, a framework that tackles the completion task whilst striving to retrieve healthy recipes. While in previous studies the relationships between ingredients are investigated and result in ingredient recommendations for the purpose of list-completion or substitution [5, 7], here we aim to generate full healthy recipe recommendations. Specifically, we first propose an embedding-based ingredient predictor to predict the relevant ingredients with some given ingredients, and then utilize a multi-layer perceptron-based predictor to capture the quantities of supplemented ingredients. Using an iterative process, with nutritional compatibility considered at each step, we generate a healthy pseudo-recipe with a set of ingredients and the corresponding amounts. This pseudo-recipe is then matched against the recipes in the dataset to find similar healthy recommendations. The datasets used in this paper are composed of real-world online recipes collected from two major food websites, namely Allrecipes.com [8] and Yummly.com [9]. The contributions of this paper are summarized as follows.

- We propose a healthy recipe recommendation framework (NutRec), which first builds a healthy pseudo-recipe considering the nutritional values and then scans the recipe dataset for items resembling the pseudo-recipe. Our proposed NutRec relies not only on the relations between the ingredients themselves but also on those of their quantities, which ultimately dictate the healthiness of a recipe. To the best of our knowledge, no prior study has incorporated these features.
- The pseudo-recipe is a list of ingredients with their quantities, and the nutritional values of the pseudo-recipe should match the predefined targets as best as possible. To generate the pseudo-recipe, we first propose an embedding-based ingredient predictor, which embeds all the ingredients into a latent space and predicts the supplemented ingredients based on the distances of ingredient representations; we then propose an amount

predictor to compute the quantities of the supplemented ingredients.

- We conduct extensive experiments with two real recipe datasets, and the experimental results confirm the superiority of our methods over the baselines. To facilitate the community research, we have publicly released the datasets <sup>1</sup>.

The rest of the paper is structured as follows. Section 2 presents prior research done in the field, and Section 3 introduces our proposed framework NutRec. In Section 4 we describe the evaluation methods and discuss the attained results, and finally in Section 5 we conclude the findings and propose future endeavors.

## 2. Related Work

Recently, recommendation has been a hot research problem with the booming of social networks. Scholars have carried out many research works on different fields [10, 11, 12, 13, 14], e.g., location recommendation, movie recommendation. In this paper, we focus on recipe recommendation and discuss the recent progress of the related methods and applications.

### 2.1. Recipe recommendation

Recipe recommendation is a sub-domain of the larger food recommendation task, and the problem of recommending healthy recipes has been tackled in numerous studies. In literature, *healthiness* is measured by comparing macro-nutrient values against several international guidelines. The World Health Organization (WHO) and the United Kingdom Food Service Agency (FSA) guidelines are adopted in [4, 15, 16]. Furthermore, the Nutrient Reference Values of Australia and New Zealand (NRV) is incorporated in [17]. Within the realm of healthy recipe recommenders, the datasets explored vary between pre-existing healthy recipes ([6, 18, 19, 20]) and online recipe datasets ([4, 15, 17]). Online

---

<sup>1</sup><https://github.com/mchenSDU/NutRec>

data repositories pose a greater challenge as they hold an inherent tendency for unhealthiness. In what follows, we summarize the related recommendation methods according to the data types.

### *2.1.1. Recommendation with pre-existing healthy recipes*

Traditionally, recommendation methods rely on ratings and strive to suggest recipes the user would rate high. For example, Freyne and Berkovsky [6] use recipe ratings to explore the relationship between healthy recipes and their ingredients. They devise a break-down technique and examine the feasibility of extracting the users' ingredient preferences from their ratings on recipes. They demonstrate their method can surpass a collaborative filtering baseline approach. However, when dealing with online recipes, users' ratings on recipes are not good indicators for healthiness, as many recipes with high ratings are pretty unhealthy.

Furthermore, some methods rely on domain experts' advice and are generally knowledge-based. Fudholi et al. [18] design an ontology-based daily menu assistance system to meet daily energy requirements. Espin et al. [19] present a nutritional recommender system, helping elderly users to draw up their own healthy diet plans following the nutritional experts' guidelines. Tumnark et al. [20] describe a food and nutrition ontology working with a rule-based knowledge framework to satisfy the specific dietary needs of weightlifters. These studies use predefined diets and menus which are all considered appropriate but not necessarily compatible with the user's needs. The recommender component in those systems may accept, reject or adjust the menus based on rules set by nutrition experts or clinical guidelines. Unfortunately, the advantages of this methodology cannot be effortlessly applied to online recipes. Online data are not guaranteed to include all required ontology properties nor are all recipes valid candidates (i.e. being healthy).

### *2.1.2. Recommendation with online recipe datasets*

To improve the healthiness of the recommendations, multiple methods are presented with online recipe datasets. For instance, Trattner and Elsweiler [4] apply post-filtering on the generated recommendations and re-rank them to make the healthier recipes appear on the top of the prediction list. Seeing that online repositories often contain multiple versions of the same dish, Elsweiler and Morgan [17] propose a solution which diverges from the quest to find a single healthy recipe and instead aims to generate daily meal plans. These plans are balanced recipe combinations which are obtained via exhaustive search. Further, they [15] evaluate the possibility of substituting one recipe with a similar yet healthier alternative. In addition, Trattner et al. [21] analyze the upload behavior of an online food community platform, and find that social connections in the form of online friendship relations bear a great potential to predict what types of recipes user are going to create in the future and which ingredients are going to be used. The research contributes to a better understanding in online food behavior and benefits healthy recipe recommendation.

In this paper, we focus on online recipes and try to tackle the challenge from another angle. Our method first generates a healthy recipe-like draft and then searches the dataset for matchings. Unlike other papers, it does not rely on ratings nor on the availability of nutritional information for the recipes.

### *2.2. Recipe completion*

Recipe completion requires an understanding of the relations between ingredients in a recipe. A recent extensive survey performs such an investigation with the aim of characterizing different cuisines [1]. Kikuchi et al. [22] demonstrate the effects of seasonality on recipes' content and illustrate how ingredient co-appearance in recipes changes throughout a year. In [23] the typicality of ingredients within a recipe category (e.g. "hamburger") is defined. This measure is used to help determine how well a specific ingredient fits into a target recipe category.

Table 1: Notations and descriptions.

Notations	Descriptions
$r, i$	recipe, ingredient
$N_r$	the number of ingredients in recipe $r$
$\mathbf{v}_i$	the embedding vector of target ingredient $i$
$\mathbf{c}_i$	the contextual vector of target ingredient $i$
$\mathbf{v}'_i$	the embedding vector of contextual ingredient $i$
$d$	the dimension of latent space
$N$	the set of macro-nutrients
$t$	the macro-nutrients' target values
$p$	the predicted macro-nutrients' values
$n$	the number of ingredients adding to the pseudo-recipe
$k$	the number of recommended healthy recipes

The two most relevant works to this paper are [5] and [7]. The *complementary network*, introduced in [7], models all of the recipes' ingredients as vertices of a graph. The weights between any ingredient pair represent the probability of co-appearance. To extract insights from this model they use clustering algorithms on the network. Even though they set out to solve different problems, e.g. substituting ingredients in recipes, this network model lends itself nicely to a completion task. Out of all the related papers, [5] is the only one to intently deal with the completion task. The tools they employ are matrix factorization (*MF-based*) models, which have two major components including non-negative matrix factorization and two-step least root squares. These methods are able to extract compatibility information from the recipe-ingredient matrix.

### 3. Proposed Framework: NutRec

#### 3.1. General framework

The proposed framework NutRec is inspired by a human-like deliberation and seeks to generate healthy recommendations over an online non-healthy

dataset. The user’s preferences are accommodated in the form of an ingredient-list and are passed to the framework as input. NutRec consists of three main stages: 1) predicting the relevant ingredients with an ingredient-list; 2) predicting the amounts of the relevant ingredients; 3) creating a healthy pseudo-recipe and searching for similar recipes in the dataset. The working process of NutRec is presented in full in Algorithm 1. Tab. 1 lists the notations used in this paper.

- **Ingredient predictor.** We propose an embedding-based predictor to find candidate relevant ingredients by computing top compatible ingredient lists with the current ingredient-set *prSet* (line 3).
- **Amount predictor.** A list of ingredients does not suffice to calculate nutritional values, to accomplish this task we demand the rations. We develop an amount predictor which enables us to predict the quantities of a given ingredient-set (line 4).
- **Healthy recipe recommendation.** By using the ingredient predictor and the amount predictor, the *FindBestIngredientToAdd* assess how each candidate ingredient impacts the nutritional values and returns the best ingredient to append to the working set. We iteratively execute this process for  $n$  times, and generate a healthy pseudo-recipe, i.e. a data-structure containing the ingredients along with their respective amounts (line 9). Finally, with the healthy pseudo-recipe, we search the dataset for the top- $k$  most similar recipes (line 10).

### 3.2. The ingredient predictor

#### 3.2.1. model description

In order to retrieve the most befitting ingredients, we propose an embedding-based ingredient predictor (IP-embedding), in which the ingredients are projected into a latent space and the ones that usually occur in a recipe are close to each other. Inspired by the recent progress of deep learning and neural networks [24, 25, 26], we propose to use a distributed representation method to model the

---

**Algorithm 1** NutRec: Recommend

---

**Input:**  $s$ : the initial ingredient-set $k$ : the number of recommended healthy recipes**Output:**  $l$ : the list of recommended recipes**Tuning params:**  $n$ : the number of ingredients adding to the pseudo-recipe $cos$ : cosine-weight for similar recipes search

```
1:  $prSet = s$ 
2: for  $i = 0$  to  $n$  do
3:    $candidateIngredients = \text{IngredientPredictor}(prSet)$ 
4:    $amountofCandidateIngredients = \text{AmountPredictor}(prSet)$ 
5:    $bestIngredient = \text{FindBestIngredientToAdd}(candidateIngredients,$ 
6:      $amountofCandidateIngredients)$ 
7:    $prSet += bestIngredient$ 
8: end for
9:  $pseudoRecipe = \text{CreateRecipeFromSet}(prSet)$ 
10:  $l = \text{FindSimilarRecipes}(pseudoRecipe, cos, k)$ 
11: return  $l$ 
```

---

generation of the given recipe. Given a recipe  $r : \{i_1, i_2, \dots, i_{N_r}\}$  containing  $N_r$  ingredients, the objective function is to maximize the probability of each target ingredient  $i_a$  given its corresponding context information:

$$g(i_a) = Pr(i_a | Context(i_a)), \quad (1)$$

where we consider the ingredients except  $i_a$  in the recipe  $r$  as its context.

Formally, we model each target ingredient  $i_a$  with a  $d$ -dimensional embedding vector  $\mathbf{v}_{i_a} \in \mathbb{R}^d$  and embed each contextual ingredient into the same latent vector space. Given an ingredient  $i_a$ , we average the embedding vectors of multiple contextual ingredients to compute a contextual vector  $\mathbf{c}_{i_a}$ ,

$$\mathbf{c}_{i_a} = \frac{1}{N_r - 1} \sum_{t=1, t \neq a}^{N_r} \mathbf{v}'_{i_t}, \quad (2)$$

where  $\mathbf{v}'_{i_t}$  is the embedding vector of the contextual ingredient  $i_t$ . Note that, we

distinguish between the role of a target ingredient and a contextual ingredient, and represent the same ingredient  $i_a$  using two different vectors ( $\mathbf{v}_{i_a}$  and  $\mathbf{v}'_{i_a}$ ) depending on which role it takes. We then apply a softmax function to compute the probability of a target ingredient  $i_a$  given its contextual vector as follows:

$$Pr(i_a|Context(i_a)) = \frac{\exp(\mathbf{c}_{i_a}^T \cdot \mathbf{v}_{i_a})}{\sum_{i \in \mathcal{I}} \exp(\mathbf{c}_{i_a}^T \cdot \mathbf{v}_i)}, \quad (3)$$

where  $\mathcal{I}$  is the set of ingredients. Such a model connects the target ingredient and its contexts via the embedding representations.

For parameter learning, we need to maximize the log probability defined in Eq. (1) over all the recipes. However, directly optimizing this objective function is impractical because the cost of computing the full softmax for the multiclassifier to predict the target ingredient is extremely high. Therefore, we adopt the efficient and effective negative sampling strategy [27, 28] to approximate the full softmax. When we train the vector of  $i_a$ , we first obtain a corresponding negative sample set  $NEG(i_a)$ , in which  $i_x$  is not the same as  $i_a$  if  $i_x \in NEG(i_a)$ . Then we define

$$L^{i_a}(i) = \begin{cases} 1, & i = i_a; \\ 0, & i \neq i_a, \end{cases} \quad (4)$$

where  $L^{i_a}(i)$  is the label of the ingredient  $i$ . The labels of positive instances are 1, and 0 otherwise.

Given the contextual ingredients of a target ingredient  $i_a$ , we want to maximize the occurrence probability of  $i_a$  and meanwhile minimize the occurrence probability of negative samples  $i_x \in NEG(i_a)$ . The objective function for the target ingredient  $i_a$  therefore becomes

$$g(i_a) = \prod_{i \in i_a \cup NEG(i_a)} Pr(i|Context(i_a)), \quad (5)$$

where

$$Pr(i|Context(i_a)) = \begin{cases} \sigma(\mathbf{c}_{i_a}^T \mathbf{v}_i), & L^{i_a}(i) = 1; \\ 1 - \sigma(\mathbf{c}_{i_a}^T \mathbf{v}_i), & L^{i_a}(i) = 0, \end{cases} \quad (6)$$

in which  $\sigma(z) = (1 + \exp(-z))^{-1}$  is the sigmoid function.

Further,

$$\begin{aligned}
g(i_a) &= \prod_{i \in i_a \cup NEG(i_a)} Pr(i|Context(i_a)), \\
&= \prod_{i \in i_a \cup NEG(i_a)} [\sigma(\mathbf{c}_{i_a}^T \mathbf{v}_i)]^{L^{i_a}(i)} \cdot [1 - \sigma(\mathbf{c}_{i_a}^T \mathbf{v}_i)]^{1-L^{i_a}(i)}, \\
&= \sigma(\mathbf{c}_{i_a}^T \mathbf{v}_{i_a}) \prod_{i \in NEG(i_a)} (1 - \sigma(\mathbf{c}_{i_a}^T \mathbf{v}_i)).
\end{aligned} \tag{7}$$

Finally, we define the object function for all the ingredients,

$$\begin{aligned}
\mathcal{L} &= \sum_{r \in \mathcal{R}} \sum_{a=1}^{N_r} \log g(i_a), \\
&= \sum_{r \in \mathcal{R}} \sum_{a=1}^{N_r} \left( \log[\sigma(\mathbf{c}_{i_a}^T \mathbf{v}_{i_a})] + \sum_{i \in NEG(i_a)} \log[1 - \sigma(\mathbf{c}_{i_a}^T \mathbf{v}_i)] \right), \\
&= \sum_{r \in \mathcal{R}} \sum_{a=1}^{N_r} \left( \log[\sigma(\mathbf{c}_{i_a}^T \mathbf{v}_{i_a})] + \sum_{i \in NEG(i_a)} \log[\sigma(-\mathbf{c}_{i_a}^T \mathbf{v}_i)] \right).
\end{aligned} \tag{8}$$

Given an incomplete recipe, we regard all its ingredients as the context, and compute  $Pr(i|Context(i))$  based on Eq. (3) for each candidate ingredient  $i$ , and return the top- $N_i$  ingredients as outputs.

### 3.2.2. Parameter learning

IP-embedding needs to learn the embedding vectors of all the ingredients. We represent all the parameters with  $\Theta$ , and learn the IP-embedding model by using maximum a posterior (MAP):

$$\Theta = \arg \max \sum_{r \in \mathcal{R}} \sum_{a=1}^{N_r} \log g(i_a) - \frac{1}{2} \lambda \|\Theta\|^2, \tag{9}$$

where  $\lambda \|\Theta\|^2$  is the regularization term.

All parameters are trained using the stochastic gradient ascent method. During the training process, the algorithm iterates over the target ingredients of all the recipes. At each time, a target ingredient  $i_a$  with its contexts is used for update. For a certain context  $Context(i_a)$ , the target ingredient  $i_a$  is a positive sample, and we randomly sample  $N_e$  unobserved ingredients as negative

samples. After computing the loss function, the error gradient is obtained via backpropagation and we use the gradient to update the parameters in our model.

$$\begin{aligned}
\mathbf{v}_{i_a} &= \mathbf{v}_{i_a} + \eta ([1 - \sigma(\mathbf{c}_{i_a}^T \mathbf{v}_{i_a})] \cdot \mathbf{c}_{i_a} - \lambda \mathbf{v}_{i_a}), \\
\mathbf{v}_{i_x} &= \mathbf{v}_{i_x} + \eta (-\sigma(\mathbf{c}_{i_a}^T \mathbf{v}_{i_x}) \cdot \mathbf{c}_{i_a} - \lambda \mathbf{v}_{i_x}), i_x \in NEG(i_a), \\
\mathbf{v}'_{i_c} &= \mathbf{v}'_{i_c} + \eta ([1 - \sigma(\mathbf{c}_{i_a}^T \mathbf{v}_{i_a})] \cdot \mathbf{v}_{i_a} - \sum_{i_x \in NEG(i_a)} [\sigma(\mathbf{c}_{i_a}^T \mathbf{v}_{i_x}) \cdot \mathbf{v}_{i_x}] - \lambda \mathbf{v}'_{i_c}), \\
i_c &\in Context(i_a),
\end{aligned} \tag{10}$$

where  $\eta$  is the learning rate.

### 3.3. The amount predictor

This component is tasked with predicting the gram-amounts a set of ingredients are likely to be used in a recipe. When evaluating the nutritional values of an ingredient-set, we need to know the proportions among the ingredients. Since a programmatic approach to this problem is likely to result in non-realistic values (e.g., a kilo of onions and 10 grams of flour), to gain intuition about real-world ratios we have built an amount predictor.

The amount predictor is based on a dense one hidden layer network with ReLU activation function. The input and output layers' dimensions equate to the number of unique ingredients in the dataset ( $m$ ), and the optimal size for the hidden layer is determined by tuning. The input for the network is a binary vector  $\vec{x}$  of length  $m$  with the value 1 assigned to the ingredients which are included in the set and 0 to the rest. The output of the network is a numerical vector  $\vec{x}'$  of the same size with the amount values (in grams) for all ingredients.

The amount predictor is trained using the Keras framework<sup>2</sup>. A binary recipe-ingredient matrix  $X$  is utilized as the input along with an additional numerical matrix  $X'$  of recipe-ingredient-amounts which serves as the target.

---

<sup>2</sup>F. Chollet, 2015. Keras. <https://keras.io>

### 3.4. Healthy recipe recommendation

#### 3.4.1. Finding the best supplemented ingredients and generating the pseudo-recipe

In order to effectively capture the user’s preference and maintain the healthiness, we evaluate the content similarity between the target nutrients and each candidate set. In particular, Mean Squared Errors ( $MSE$ ) measure is adopted as the distance measure. The error in this scenario is the distance between the target macro-nutrient values and the predicted ones. Formally:

$$MSE = \frac{1}{|N|} \sum_{i \in N} (t_i - p_i)^2, \quad (11)$$

where  $N$  is the set of macro-nutrients (protein, carbohydrates, sugars, fat, saturated fat, sodium and fiber),  $t$  is the macro-nutrients target values and  $p$  is the predicted macro-nutrients values of the current set extended by the inspected candidate. The ingredient which results in the lowest MSE score is chosen.

The process halts after  $n$  iterations or when no new ingredient can improve upon the error. In this work we have limited the number of iteration to 5, so as to not exceed the average number of recipe ingredients (as we will show in Tab. 2). Additionally, we do not wish to encourage the algorithm to pick ingredients which typically appear in small amounts (and thus do not meaningfully influence the MSE) just for the sake of adding more ingredients. The final product of this step is our pseudo-recipe.

#### 3.4.2. Retrieving similar recipes

We point out our aspiration is to find recipes which resemble the pseudo-recipe both in the ingredients themselves but also their quantities. Unfortunately, these two properties do not necessarily coincide. To mitigate this issue, we prescribe *similarity* as a weighted average of the Jaccard and cosine metrics:

$$\text{sim} = \text{COS}_{\text{weight}} \cdot \text{sim}_{\text{cos}} + (1 - \text{COS}_{\text{weight}}) \cdot \text{sim}_{\text{jaccard}} \quad (12)$$

Cosine similarity captures the likeness in terms of amounts:

$$\text{sim}_{\text{cos}} = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} = \frac{\sum_{j=1}^m \vec{a}_j \cdot \vec{b}_j}{\sqrt{\sum_{j=1}^m \vec{a}_j^2} \sqrt{\sum_{j=1}^m \vec{b}_j^2}}, \quad (13)$$

where  $\vec{a}$  is the pseudo-recipe amounts vector and  $\vec{b}$  is a recipe amounts vector, both of length  $m$  (the number of unique ingredients in the dataset). Jaccard index is used to emphasize shared ingredients:

$$\text{sim}_{\text{jaccard}} = \frac{|A \cap B|}{|A \cup B|}. \quad (14)$$

Here  $A$  and  $B$  are the ingredient-sets of the pseudo-recipe and an examined recipe, respectively.

## 4. Experimental Results

### 4.1. Datasets

The datasets utilized in this paper are collected from Allrecipes<sup>3</sup> and Yummly<sup>4</sup>. All nutritional values for the recipes are available on the websites. The ingredients and amounts are parsed in a semiautomatic manner, using fuzzy matching between the recipes’ ingredients and the foods in the USDA dataset<sup>5</sup>. About 80% of ingredients are recognized and at least 70% are parsed. A parsed ingredient is a recognized ingredient which we are able to quantify in grams. To make our models more robust, we filter those recipes that only contain one ingredient. We have publicly released our datasets to facilitate the community research<sup>6</sup>.

Tab. 2 displays the characteristics of the datasets. The WHO scores are computed using the manner described in [4], and their distribution is displayed

---

<sup>3</sup><https://www.allrecipes.com>

<sup>4</sup><https://www.yummly.com>

<sup>5</sup>US Department of Agriculture, Agricultural Research Service, Nutrient Data Laboratory. USDA National Nutrient Database for Standard Reference, Release 28. Version Current:

September 2015. Available: <http://www.ars.usda.gov/ba/bhnrc/ndl>

<sup>6</sup><https://github.com/mchenSDU/NutRec>

Table 2: Data statistics

	Allrecipes	Yummly
Number of recipes	36,429	89,413
Average no. of ingredients per recipe	7.2	7.6
Recognized ingredients	82%	81%
Parsed ingredients	77%	71%
Unique ingredients	465	512

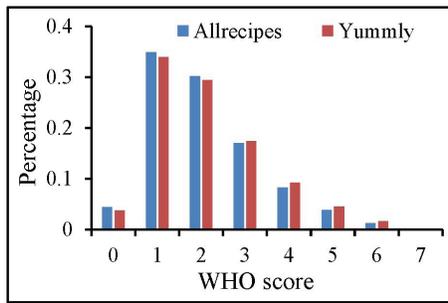


Figure 1: WHO score distribution in Allrecipes and Yummly datasets. WHO score reflects how well a recipe meets the WHO guideline on the scale of 0-7 (poorly-well). A recipe with a high WHO score is considered healthy.

in Fig. 1. Since healthy recipes entail high WHO scores, it can be clearly seen that most recipes in the datasets are unhealthy. Lastly, nutrition information for ingredients is incorporated via the USDA dataset.

For both datasets, we randomly split them into three collections in proportion of 8:1:1 as the training set, validation set, and test set, and perform 5 runs (with the same data split) to report the average of the results. All the experiments are done on a 3.4GHz Intel Core i7 PC with 16GB main memory.

#### 4.2. Evaluation on the ingredient predictor

The proposed ingredient predictor could be applied in the task of recipe completion and is examined by the following method. For each recipe, a random ingredient is removed and the ability of the models to retrieve it is tested. This

method is also employed for our baseline predictors [5, 29]. With IP-embedding, we first build a contextual vector  $c_i$  based on Eq. (2), and then compute  $Pr(i|c_i)$  for each candidate ingredient  $i \in \mathcal{I}$  based on the following function:

$$Pr(i|c_i) \propto \mathbf{c}_i^T \cdot \mathbf{v}_i. \quad (15)$$

Finally, we predict the missing ingredient according to the probability. The results of this evaluation are reported as:

- Percent of predictions with the removed ingredient ranked  $\leq 10$ ,
- Mean rank of the removed ingredient,
- Median rank of the removed ingredient.

An optimal predictor would return the missing ingredient at the top of the list. Here we consider it a good prediction if the removed item appears within the first 10 results. Therefore, we would like the mean and median ranks to be low and the percent of good predictions to be as high as possible.

#### 4.2.1. Baselines

We choose three methods as the baselines and compare our IP-embedding with them.

- **Graph-based ingredient predictor (IP-graph)** is a graph-based model  $G(V, E)$ , where  $V$  consists of all the unique ingredients. Edges  $E$  connect any two ingredients that co-appear in at least one recipe and the edge weights are the co-appearances' counts. IP-graph returns the top- $N_i$  candidate ingredients from a vertex list sorted by the combined edge weights [29].
- **Multi-layer perceptron-based ingredient predictor (IP-MLP)** uses the same network structure as that of the proposed amount predictor. The output of the amount predictor is a numerical vector  $\vec{x}'$  of the same size with the amount values (in grams) for all ingredients. IP-MLP returns the top- $N_i$  ingredients with the highest values in  $\vec{x}'$  as the candidate ingredients.

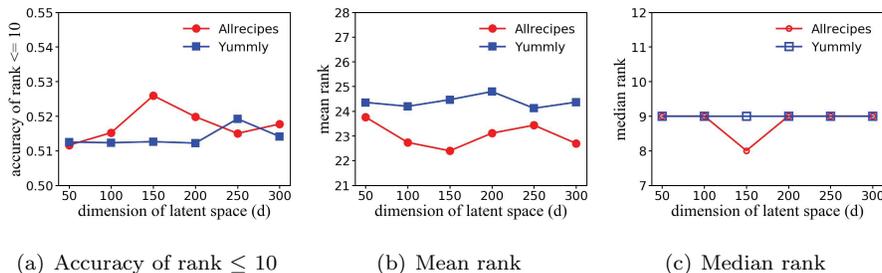


Figure 2: Empirical performance of tuning the dimension of latent space (IP-embedding).

- **Nonnegative matrix factorization-based ingredient predictor (IP-NMF)** is inspired by the nonnegative matrix factorization model originally described in [5]. IP-NMF decomposes the recipe-ingredient matrix into two smaller matrices  $W$  and  $H$  with dimensions  $u \times c$  and  $c \times m$ , where  $c$  is the number of latent factors,  $u$  is the number of recipes and  $m$  is the number of unique ingredients. Further, it computes a resulting vector with  $W$  and  $H$ , and returns the top- $N_i$  predictions correspond to ingredients with the highest values in the vector.

#### 4.2.2. Parameter setting and tuning

We measure the effect of the dimension of latent space  $d$  for IP-embedding. We increase  $d$  from 50 to 300 with a step of 50, and demonstrate the results in Fig. 2. The performance of IP-embedding changes little with each  $d$ . When  $d$  is equal to 150, IP-embedding performs the best on the Allrecipes dataset, and IP-embedding reaches the optimal when  $d$  is 250 on the Yummly dataset. The optimal parameter is used in all subsequent experiments.

As a side contribution, IP-embedding could embed the ingredients into a latent space, and a pair of ingredients with more relatedness should be projected into closer embedding vectors. To validate this semantic correlation, we list the top-10 similar ingredients for a given ingredient. Here we choose “pasta”, “beef”, and “vegetable oil” as the examples of query ingredient, and return the top-10 similar ingredients based on the cosine similarity between the embedding vectors, as shown in Tab. 3.

Table 3: Similar ingredients discovered by IP-embedding with a given ingredient

	<b>pasta</b>	<b>beef</b>	<b>vegetable oil</b>
top-10 similar ingredients	spaghetti	venison	canola oil
	bagels	mutton	olive oil
	cheese ravioli	lean ground beef	lard
	noodles	ground turkey	oil
	bologna	beef sausage	coconut oil
	vermicelli	turkey sausage	large carrots
	hearts of palm	roast beef	salted butter
	asparagus	rolls	margarine
	artichoke hearts	breakfast sausage	finely chopped onion
	rolls	steak	eggplant

- Given the query ingredient “pasta”, the most similar ingredient is “spaghetti”, which is the same food as “pasta”. Furthermore, the top-10 similar ingredients of “pasta” are mainly cooked wheaten food.
- The top-10 similar ingredients of “beef” are all meat or meat sausage, e.g., “venison”, “mutton”, and “beef sausage”.
- The top-5 similar ingredients of “vegetable oil” contain multiple kinds of oil, e.g., “canola oil”, “olive oil”, and “lard”.

We can see that the similar ingredients discovered by IP-embedding demonstrate unique characteristics about the query ingredient in an explainable manner. Therefore, the learned vectors of ingredients include the semantic correlation and are pretty valuable for many tasks, e.g., ingredient clustering.

#### 4.2.3. Comparison with existing methods

Finally, we compare our proposed IP-embedding with these baselines. For IP-NMF, we test it with 2-4 components (i.e., the number of hidden factors). IP-NMF obtains the best performance when the number of hidden factors is

Table 4: Performances of the four ingredient predictors

		IP-graph	IP-MLP	IP-NMF	<b>IP-embedding</b>
<b>Allrecipes</b>	Rank $\leq$ 10	45.5%	31.0%	49.5%	<b>52.6%</b>
	Mean rank	29.6	44.9	37.2	<b>22.4</b>
	Median rank	12	27	11	<b>8</b>
<b>Yummly</b>	Rank $\leq$ 10	43.5%	28.0%	50.0%	<b>51.9%</b>
	Mean rank	32.4	49.8	37.4	<b>24.1</b>
	Median rank	13	30	11	<b>9</b>

2. For IP-MLP, we set the size of hidden layer at 128, 256, 512, respectively, and find that IP-MLP performs the best when the size of hidden layer is 128. We summarize the results of the four ingredient predictors in Tab. 4. The best results are highlighted in boldface. IP-graph constructs a graph based on the recipes and it obtains the decent performance. IP-MLP builds a simple network and it predicts the missing ingredient based on the predicted amount of each candidate ingredient. IP-MLP performs the worst as it leverages the indirect information. IP-NMF predicts the missing ingredient based on non-negative matrix factorization, and it performs better than IP-graph and IP-MLP in terms of accuracy of rank  $\leq$  10. Our proposed IP-embedding produces the best results out of the four models, as it models the relation between the target ingredient and its contexts, and predicts the missing ingredient according to the context information. It is able to include the correct ingredient within the top-10 elements in 52.6% of the cases on the Allrecipes dataset and 51.9% of the cases on the Yummly dataset. Its mean and median scores are also lower than the competitors. Note that, the improvements of our IP-embedding over all the baselines are statistically significant in terms of paired t-test [30] with p value  $< 0.01$ .

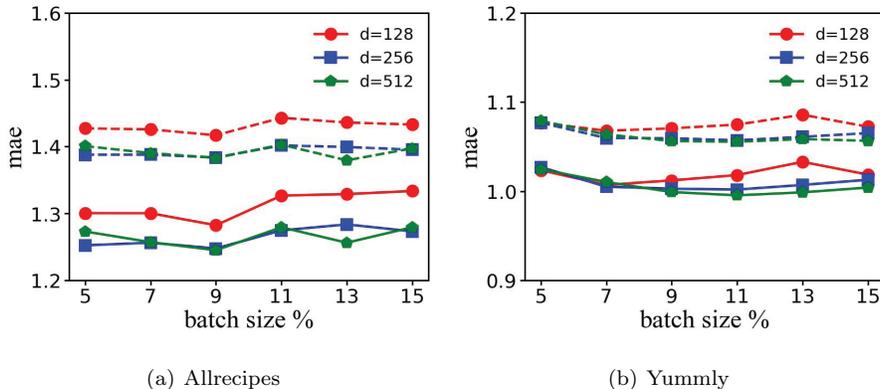


Figure 3: The performance of the amount predictor for Allrecipes and Yummly. Solid lines represent the mae on the training set and dashed lines represent the mae on the validation set.

#### 4.3. Evaluation on the amount predictor

The amount predictor is evaluated by the mean absolute error (mae). The evaluation is performed on the training set and the validation set while varying the size of the hidden layer and the batch size. Fig. 3 illustrates these results. A batch size of 9% produces the best results on both datasets. The lowest train and validation mae scores for Allrecipes are 1.25 and 1.38 respectively with a hidden layer dimension of 512. For Yummly the lowest mae scores obtained are 1.0 for the training set and 1.06 on the validation set with a hidden layer the size of 512 neurons. It is not surprising a lower validation mae is observed on Yummly as its size provides more training opportunities.

#### 4.4. Evaluation on the NutRec

As no state-of-the-art method for comparison is available, we formulate an alternative design and validate whether our proposal could recommend healthier recipes through empirical analysis. The evaluation of the framework NutRec is performed using 120 most frequent item sets. Frequent sets are sets of ingredients which repeatedly appear together in recipes. The parameters are set to  $n=5$  and  $k=10$ . Meaning, NutRec increases the initial ingredient set by 5 ingredients and returns the top-10 most similar healthy recipes.

Table 5: Target macro-nutrients

Nutrient	WHO guideline	Target <sup>a</sup>
Protein	10%-15%	12.5% = 62.5g
Carbohydrates	55%-75%	65% = 325g
Sugar	< 10%	6.6% = 33.3g
Fat	15%-30%	22.5% = 50g
Saturated fat	< 10%	6.6% = 14.8g
Sodium	< 2g	1.33g
Fiber	> 25g	37.5g

<sup>a</sup>Percentage conversion is based on 2000 Kcal/day diet.

To measure the performance of NutRec, we adopt the mean WHO score, which is based on the recommendations’ WHO score distribution. We compare our NutRec with a *random-draw* baseline. The random-draw is computed using all recipes containing the initial sets. We average over these recipes’ WHO scores to represent the baseline. In addition to the mean WHO score, we have studied another error-based metric, using logic similar to Eq. (11), in our preliminary experiments. However, our tests proved this measure to be inappropriate for precise targets. This is due to the fact that even though wide gaps between the macro-nutrient targets and the actual recipe values induce large errors, a combination of minor deviations can have the same effect.

The target macro-nutrient values are set as detailed in Tab. 5. The table displays the seven macro-nutrients along with their recommended portion of daily energy intake. The targets are expressed in grams. Whenever a range is clearly defined we use its mean value. For upper or lower bounds (such as the case with sugar, sodium and fiber) we use a 1.5 multiplier. The target values are based on a 2000 Kcal intake and a conventional conversion is used: a single gram of protein, carbohydrates or sugar amounts to 4 Kcal and one gram of fat or saturated fat is equal to 9 Kcal.

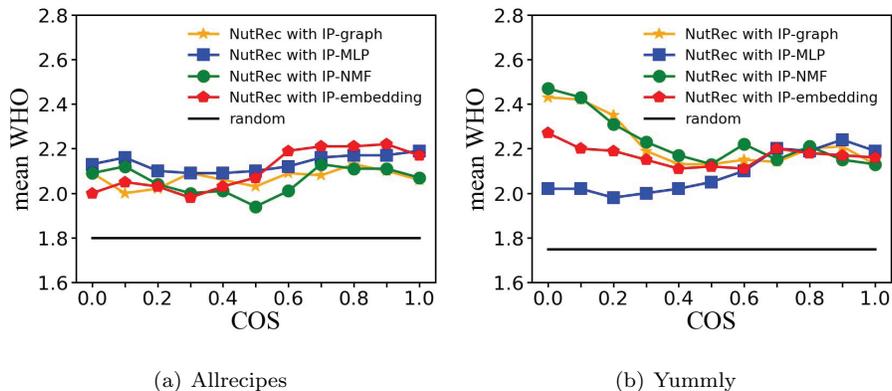


Figure 4: Mean WHO score of recommended recipes using different ingredient predictors.

We evaluate the performance of four ingredient predictors in Section 4.2.3, and we incorporate them into the framework NutRec respectively to validate the effectiveness. The mean WHO scores of the predictions as a function of  $\text{COS}_{\text{weight}}$  (COS) are demonstrated in Fig. 4. The colors correspond to the NutRec with the four ingredient predictors. IP-graph is denoted by a yellow line, IP-MLP is blue, IP-NMF is green and IP-embedding is red. The random-draw distribution is laid out by the black line. This chart illustrates that all COS settings improve the random-draw baseline, meaning our NutRec indeed recommends healthier recipes for users. For example, the baseline mean WHO scores for the top-120 most frequent sets in Allrecipes and Yummly are 1.80 and 1.75 respectively; the NutRec with IP-embedding reaches 2.22 with COS equal to 0.9 on the Allrecipes dataset and the NutRec with IP-NMF could reach 2.47 on the Yummly dataset. Additional investigation reveals that higher cosine-similarity weights actually performed slightly better for most of the cases, meaning that the amounts of ingredients play a pivotal role in the healthy recipe recommendation. Furthermore, the NutRec with our proposed IP-embedding performs relatively better, especially on the Allrecipes dataset.

Further, we explore whether the mean WHO scores can be increased by training the amount predictor using only recipes with the WHO scores greater than 3 (i.e. using a dataset with mostly healthy examples). The results are

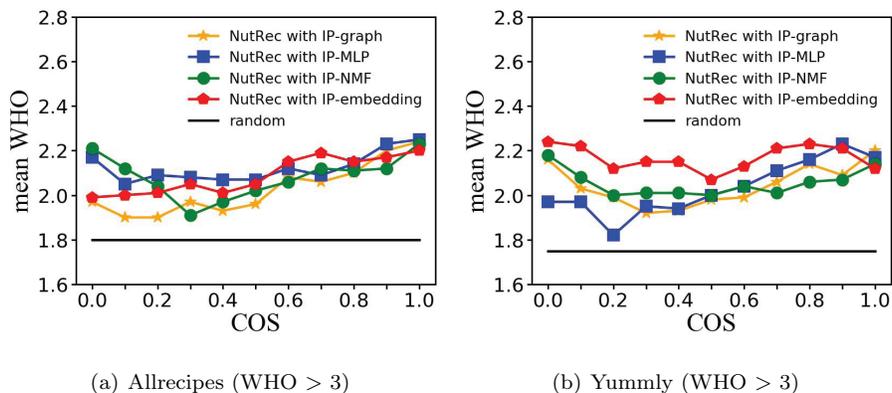


Figure 5: Mean WHO score of recommended recipes, using different ingredient predictor, when training the amounts network on recipes with WHO > 3.

illustrated in Fig. 5. The actual improvement according to the mean WHO score is negligible. In most of the cases, the mean WHO scores are lower than those in Fig. 4. We will point out two striking trends apparent in the results. First, as can be deduced from Fig. 1, imposing a minimum WHO score limit drastically cuts down the training data and subsequently impairs the model’s predictive power. It is plausible the model’s accuracy deterioration leads to incorrect ingredient combinations, for instance the algorithm may produce an ingredient-set often found in unhealthy recipes. This is evidenced in Fig. 5 by the inferior mean WHO scores for low COS values (that is when recommending recipes based on primarily similar ingredients). A second feature, which can be recognized in Fig. 5, is the positive correlation of COS and mean WHO score, unlike the vague relation seen in Fig. 4. This implies that under this setup the predicted amounts better model healthy ingredients-ratios. Notwithstanding, the fact this experiment is not able to improve the results on Yummly suggests learning using healthy recipes may be less beneficial than having a large training set.

## 5. Conclusion and Future Work

In this paper, we have proposed a framework named NutRec to tackle the healthy recipe recommendation problem. NutRec first predicts the relevant ingredients and their amounts with some user-defined initial ingredients, and then creates a healthy pseudo-recipe considering the nutritional values. Finally, NutRec searches the top similar healthy recipes based on the pseudo-recipe in the dataset. As evident from the presented results, NutRec is able to improve the average healthiness of the recommended recipes (for a top-10 task) without requiring any pre-computed nutritional information for the recipes.

In the future, we would like to advance our study on the following topics. First, one technique to improve the performance may be via an ensemble model which will combine all four ingredient predictors. Second, in this paper healthy recipes are the focal point, but as the method strives to find recipes which best match a given nutrient target, it is credible it could prove useful for other types of goals, e.g., low-fat or high-protein diets. Finally, as users' ratings are a prominent part of recommender systems, an interesting extension for this work would be investigating a fashion to integrate ratings. Once such technique is applicable, it will be possible to compare the proposed algorithm with traditional recommender methods.

## 6. Acknowledgment

This work was supported in part by the Fundamental Research Funds of Shandong University, the Natural Science Foundation of Shandong Province of China under Grant No. ZR2019BF010, the National Natural Science Foundation of China under Grant No. 61572289, and the NSERC Discovery Grants.

## References

- [1] S. Sajadmanesh, S. Jafarzadeh, S. A. Ossia, H. R. Rabiee, H. Haddadi, Y. Mejova, M. Musolesi, E. D. Cristofaro, G. Stringhini, Kissing cuisines:

- exploring worldwide culinary habits on the web, in: Proceedings of the 26th International Conference on World Wide Web Companion, 2017, pp. 1013–1021.
- [2] W. Min, B.-K. Bao, S. Mei, Y. Zhu, Y. Rui, S. Jiang, You are what you eat: Exploring rich recipe information for cross-region food analysis, *Transactions on Multimedia* 20 (4) (2018) 950–964.
- [3] J. Freyne, S. Berkovsky, G. Smith, Recipe recommendation: accuracy and reasoning, in: Proceedings of the 19th International Conference on User Modeling, Adaptation, and Personalization, Springer, 2011, pp. 99–110.
- [4] C. Trattner, D. Elswailer, Investigating the healthiness of internet-sourced recipes: implications for meal planning and recommender systems, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 489–498.
- [5] Data-driven recipe completion using machine learning methods, *Trends in Food Science and Technology* 49 (2016) 1 – 13.
- [6] J. Freyne, S. Berkovsky, Intelligent food planning: personalized recipe recommendation, in: Proceedings of the 15th International Conference on Intelligent User Interfaces, 2010, pp. 321–324.
- [7] C.-Y. Teng, Y.-R. Lin, L. A. Adamic, Recipe recommendation using ingredient networks, in: Proceedings of the 4th Annual ACM Web Science Conference, 2012, pp. 298–307.
- [8] Allrecipes, Allrecipes — food, friends, and recipe inspiration (2018).  
URL <https://www.allrecipes.com>
- [9] Yummly, Yummly: Personalized recipe recommendations and search (2018).  
URL <https://www.yummly.com/>

- [10] C. Cui, J. Shen, L. Nie, R. Hong, J. Ma, Augmented collaborative filtering for sparseness reduction in personalized poi recommendation, *ACM Transactions on Intelligent Systems and Technology* 8 (5) (2017) 71.
- [11] M. Chen, X. Yu, Y. Liu, Mpe: a mobility pattern embedding model for predicting next locations, *World Wide Web* (2018) 1–20.
- [12] M. Chen, Y. Liu, X. Yu, Nlpmm: A next location predictor with markov modeling, in: *Proceedings of the 18th Pacific Asia Conference on Knowledge Discovery and Data Mining*, May 2014, pp. 186–197.
- [13] C. Cui, H. Liu, T. Lian, L. Nie, L. Zhu, Y. Yin, Distribution-oriented aesthetics assessment with semantic-aware hybrid network, *IEEE Transactions on Multimedia*.
- [14] M. Chen, X. Yu, Y. Liu, Mining object similarity for predicting next locations, *Journal of Computer Science and Technology* 31 (2016) 649–660.
- [15] D. Elswailer, C. Trattner, M. Harvey, Exploiting food choice biases for healthier recipe recommendation, in: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 575–584.
- [16] S. Howard, J. Adams, M. White, Nutritional content of supermarket ready meals and recipes by television chefs in the united kingdom: cross sectional study, *Bmj* 345 (2012) e7607.
- [17] D. Elswailer, M. Harvey, Towards automatic meal plan recommendations for balanced nutrition, in: *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 313–316.
- [18] D. H. Fudholi, N. Maneerat, R. Varakulsiripunth, Ontology-based daily menu assistance system, in: *Proceedings of the 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Vol. 02, 2009, pp. 694–697.

- [19] V. Espín, M. V. Hurtado, M. Noguera, Nutrition for elder care: a nutritional semantic recommender system for the elderly, *Expert Systems* 33 (2) (2016) 201–210.
- [20] P. Tumnark, F. A. d. Conceição, J. P. Vilas-Boas, L. Oliveira, P. Cardoso, J. Cabral, N. Santibutr, Ontology-based personalized dietary recommendation for weightlifting, in: *Proceedings of the 2013 International Workshop on Computer Science in Sports*, Atlantis Press, 2013, pp. 44–49.
- [21] C. Trattner, K. Tomasz, N. Kjetil, Investigating and predicting online food recipe upload behavior, *Information Processing & Management* 56 (3) (2019) 654–673.
- [22] Y. Kikuchi, M. Kumano, M. Kimura, Analyzing dynamical activities of co-occurrence patterns for cooking ingredients, in: *Proceedings of the 2017 IEEE International Conference on Data Mining Workshops*, 2017, pp. 17–24.
- [23] S. Yokoi, K. Doman, T. Hirayama, I. Ide, D. Deguchi, H. Murase, Typicality analysis of the combination of ingredients in a cooking recipe for assisting the arrangement of ingredients, in: *Proceedings of the 2015 IEEE International Conference on Multimedia & Expo Workshops*, 2015, pp. 1–6.
- [24] F. C. Fernández-Reyes, J. Hermsillo-Valadez, M. Montes-y Gómez, A prospect-guided global query expansion strategy using word embeddings, *Information Processing & Management* 54 (1) (2018) 1–13.
- [25] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, *Advances in Neural Information Processing Systems* 26 (2013) 3111–3119.
- [26] E. Bagheri, F. Ensan, F. Al-Obeidat, Neural word and entity embeddings for ad hoc retrieval, *Information Processing & Management* 54 (4) (2018) 657–673.

- [27] A. Mnih, Y. W. Teh, A fast and simple algorithm for training neural probabilistic language models, arXiv preprint arXiv:1206.6426.
- [28] O. Levy, Y. Goldberg, Neural word embedding as implicit matrix factorization, *Advances in Neural Information Processing Systems* 3 (2014) 2177–2185.
- [29] E. Gorbonos, Y. Liu, C. T. Hoàng, Nutrec: nutrition oriented online recipe recommender, in: *Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE, 2018, pp. 25–32.
- [30] D. Hull, Using statistical testing in the evaluation of retrieval experiments, in: *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1993, pp. 329–338.