2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)

NutRec: Nutrition Oriented Online Recipe Recommender

Elizabeth Gorbonos*, Yang Liu†, Chính T. Hoàng‡
Department of Physics and Computer Science, Wilfrid Laurier University
Waterloo, Ontario, Canada
Email: *gorb6620@mylaurier.ca, †yangliu@wlu.ca, ‡choang@wlu.ca

***Abstract*—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 algorithm which utilizes machine-learning techniques such as neural networks and matrix factorization in order to model the interactions between ingredients and their proportions within recipes for the purpose of offering suitable recommendations. The empirical results support the method's intuition and showcase its ability to retrieve healthier recipes.**

***Index Terms*—recommender system, neural network, recipes, nutrition, health**

I. Introduction

The presence of food related content on the web has become quite prominent in recent years. Along with overwhelming volumes of images on social-media, the use of online sites as a source for recipes and culinary ideas is expanding [1]. With large quantities 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 with the issue by providing users with recipes or meal plans that accommodate their preferences.

Food recommenders, which rely on a tally to indicate people's aggregated opinion, could also be a great tool to positively influence users' dietary choices, however the nature of online content holds a challenge on this end. 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 [2]. Third, multiple aspects influence our food preferences among which are geographical and cultural affiliation. Being strongly linked to these factors, cuisines are a great mean to express group inclinations [1]. Individual preferences however should be captured in other forms.

A user's culinary inclination can be perceived as a result of her ingredient preferences or due to some other latent factors. Some works use ingredients likings, whether in the form of an explicit input [3] or by indirectly learning them from recipe ratings [4]; while others employ standard collaborative filtering approaches to uncover hidden factors [2]. 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, an algorithm 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 [3], [5], here we aim to generate full recipe recommendations. The method we propose 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. We utilize a neural network-based (*NN-based*) model to capture the characteristics of ingredient quantities. Using an iterative process, with nutritional compatibility considered at each step, the method generates a set of ingredients and respective amounts. This *pseudo-recipe* is then matched against the recipes in our dataset to find similar recommendations. The datasets used in this paper are composed of real-world online recipes collected from two major food websites, namely Allrecipes.com [6] and Yummly.com [7].

The rest of the paper is structured as follows: Section II presents prior research done in the field, Section III introduces NutRec, Section IV presents the datasets adopted for this work, in Section V we describe the evaluation methods and discuss the attained results and finally in Sections VI we propose future endeavors and conclude the findings.

II. Related Work

A. Recipe recommendation

Recipe recommendation may be seen as a sub-domain of the larger food recommendation task, and the problem of recommending healthy recipes is 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 [2], [8], these studies utilize the WHO and FSA scores as presented in [9]. In [10] the "Nutrient Reference Values of Australia and New Zealand" (NRV) is incorporated. Following previous works WHO score is employed in this paper.

978-1-5386-7325-6/18/$31.00 ©2018 IEEE
DOI 10.1109/WI.2018.0-111

Within the realm of healthy recipe recommenders, the datasets explored vary between pre-existing healthy recipes ([4], [11]–[13]) and online sourced datasets ([2], [8], [10]). Online data repositories pose a greater challenge as they hold an inherent tendency for unhealthiness.

A notable line of research is ontology-based food recomendation, and it is generally knowledge-based and relies on domain experts [14]. Several papers propose ontology-based nutrition-orientated recommenders. [11] designs a system to meet daily energy requirements, [12] focuses on proper nutrition in elderly care and [13] centers on 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 guarantied to include all required ontology properties nor are all recipes valid candidates (i.e. healthy).

Traditionally, recommender methods rely on ratings and strive to suggest recipes the user would rate high. In [2] an evaluation of collaborative filtering methods is performed while [4] uses 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 her ratings on recipes. They demonstrate their method can surpass a baseline collaborative filtering approach. However, when dealing with online sourced recipes, ratings are not a good indicator for healthiness, in fact it's quite the opposite.

To improve the healthiness of the recommendations multiple methods are suggested. In [2] a post-filtering step is applied, after the recommendations are generated they are re-ranked such that the healthier recipes, in terms of WHO and FSA scores, appear at the top of the list. Seeing that online repositories often contain multiple versions of the same dish, [8] set out to evaluate the possibility of substituting one recipe with a similar yet healthier alternative (based on FSA score). To identity potential swap candidates, they use the cosine similarity measure. [10] proposes a different 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.

In this paper we too focus our efforts on web recipes and try to tackle the challenge from yet 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.

B. Recipe completion

Recipe completion requires an understanding of the relations between ingredients in a recipe. This type of analysis is done by many papers for various purposes. A recent extensive survey preforms such an investigation with the aim of characterizing different cuisines [1]. [15] demonstrates the effects of seasonality on recipes' content and illustrates how ingredient co-appearance in recipes changes throughout a year. In [16] 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.

The two most relevant works to this paper are [3] and [5]. The *complementary network*, introduced in [5], 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, [3] is the only one to intently deal with the completion task. The tools they employ are matrix factorization (*MF-based*) models: Non-negative matrix factorization and two-step least root squares. These methods are able to extract compatibility information from the recipe-ingredient matrix. Motivated by the idea presented in [3] we include a non-negative matrix factorization model as part of our process. However, different from the previous study, in our work ingredient recommendation is not an end on its own. While [3] concludes with a list of likely ingredients, we are expanding the output into full recipes.

III. Proposed Method: NutRec

The proposed algorithm 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 method as input. NutRec consists of two main stages: 1) creating a pseudo-recipe; 2) searching for similar recipes in the dataset. We will thoroughly review the flow depicted in Fig. 1 in this section.

Our pseudo-recipe, which is the output of the first step, is essentially a list of ingredients with their quantities. This sequence should amount to a healthy baseline, i.e. the nutritional values of the pseudo-recipe should match the predefined targets as best as possible. Note that the chosen ingredients are not a random collection which happens to balance nutritionally, we hope to create a practical combination and therefore need to practice sensible ingredient pairing.

Even though the pseudo-recipe might be used by experienced cooks as it is, our aim is to provide the user with definitive recipes. To achieve this, we offer to scan the dataset for items resembling the pseudo-recipe in the second step. Here we make the assumption that alike recipes will showcase similar nutritional tendencies and are thus relevant to the user. We should remark that as evidenced by the product of the first stage, there are two factors to consider when evaluating similarity, namely ingredients and quantities. The retrieved

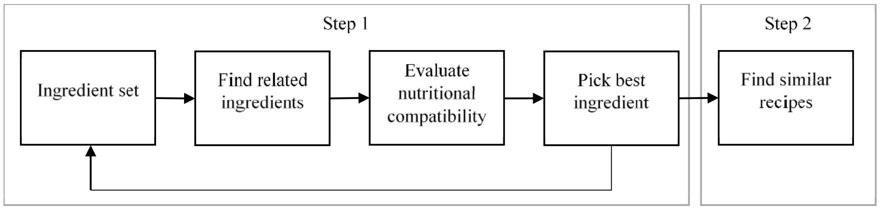


Fig. 1. NutRec flowchart

list of recipes reflects the algorithm's recommendations. The designed framework is presented in full in Algorithm 1.

For this work we adopt different solutions, including graph based, NN-based, and MF-based methods. Due to the popularity of these strategies in recent years, there might be a large array of variants along each line. We carefully select one from each which best fits our current datasets and demonstrate their effectiveness through empirical analysis. However, the methods in this framework can be easily adapted to other alternating solutions in the same vein as well.

Algorithm 1 NutRec: Recommend

```
Input: $s$: initial ingredient-set
    $k$: no. of recipes to return
Output: $l$: list of recommended recipes
Tuning params: $n$: no. of ingredients to increase $s$ by
    $ip$: ingredient predictor to utilize
    $c$: cosine-weight for similar recipes search
1: $prSet = s$
2: for $i = 0$ to $n$ do
3:     $bestIngredient$ = FindBestIngredientToAdd($prSet, ip$)
4:     $prSet$ += $bestIngredient$
5: end for
6: $pseudoRecipe$ = CreateRecipeFromSet($prSet$)
7: $l$ = FindSimilarRecipes($pseudoRecipe, c, k$)
8: return $l$
```

A. Step 1: Pseudo-recipe construction

In this phase we perform an iterative procedure where at each step the algorithm chooses one ingredient to add to the current pseudo-recipe ingredient-set $prSet$ (Algorithm 1, lines 1-5). The initial set s is provided by the user. The supplemented item, retrieved by the method *FindBestIngredientToAdd*, is required to fuse well with the ingredients in the current set in addition to balancing them nutrition-wise. Detecting the ideal ingredient calls for both a mean to find relevant candidates and a way to evaluate them. *Ingredient predictors*, which will be described hereafter, are used to introduce candidates by computing top-k compatible ingredient lists. Nonetheless, a list of ingredients does not suffice to calculate nutritional values, to accomplish this task we demand the rations. One alternative is to program an exhaustive search through all possible values, however we find drawing on existing knowledge is more useful and insightful. We devise a simple neural network model *amounts network* (*AN*) which learns ingredient interactions via their amounts. Utilizing the AN, the *amounts predictor* enables us to predict the quantities of a given ingredient-set. By using a provided ingredient predictor *ip* and the amounts predictor the function *FindBestIngredientToAdd* (Algorithm 1, line 3) assess how each candidate impacts the nutritional values and returns the best ingredient to append to the working set. Lastly, the final pseudo-recipe ingredient-set is converted into a pseudo-recipe, i.e. a data-structure containing the ingredients along with their respective amounts (Algortihm 1, line 6). This is achieved by predicting the quantities of the final set with the help of the amounts predictor.

1) Ingredient predictors: In order to retrieve the most befitting ingredients, we carefully examine, in our preliminary experiments, different factors that may affect the prediction. We will describe the three inspected ingredient prediction methods (IP1, IP2, IP3) in greater detail bellow. It is important to remark all methods incorporate individual biases. For instance, IP1 leans towards the most common ingredients (e.g. salt, pepper), while IP2 is likely to prefer ingredients which are often used in large quantities (like water) and finally IP3 derives latent factors which represent relations between ingredients. IP3 is adopted from [3] and is regarded as the baseline. We denote by n the number of recipes in the dataset and by m the number of unique ingredients in the dataset (that is within all collected recipes).

- Ingredient Predictor 1 (**IP1**) is a graph-based model $G(V, E)$ where V consists of all the unique ingredients (particularly $|V| = m$). Edges connect any two ingredients that co-appear in at least one recipe and the edge weights are the co-appearances' counts, similar to [5]. Let e be an edge connecting ingredient i_1 and i_2, then the weight of e is:

$$W(e) = \sum_{r \in \text{Recipes}} \begin{cases} 1, & \text{if } \{i_1, i_2\} \in r \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

  However different from [5] whose objective is to explore ingredient clusters and possible substitutions we aim to investigate ingredients' relations in the form of subgraphs. For this work its easy to see that the relevant ingredients are those that create a clique with the initial set's vertices, since any missing edge means no recipe contains all ingredients. The top-k ingredients are derived from a vertex list sorted by the combined edge weights.

- Ingredient Predictor 2 (**IP2**) is based on a shallow neural network. The network is designed such that given a set of ingredients it will predict the quantities those ingredients will be used in. The neural network utilized herein is a dense one hidden layer network with ReLU activation function. The input and output layers' dimensions equate to 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. Note, for ingredients whose input indicator is 0, meaning they do not participate in the set, the output may still be positive. If an ingredient which does not belong to the input set results in a high value it may imply this ingredient frequently accompanies the set in large quantities and is therefore relevant. The top-k ingredients returned by this predictor are the ingredients with the highest values in $\vec{x'}$ (for which $\vec{x}_i$=0).
- Ingredient Predictor 3 (**IP3**) is inspired by the non-negative matrix factorization (NMF) model originally described in [3]. X is the $n \times m$ binary recipe-ingredient matrix. The rows of X represent which ingredients a recipe is composed of.

  $$X_{ri} = \begin{cases} 1, & \text{if ingregient } i \text{ in recipe } r \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

  Corresponding to a parameter c, the number of latent factors, NMF decomposes X into two smaller matrices W and H with dimensions $n \times c$ and $c \times m$ respectively such that:

  $$X \approx WH \tag{3}$$

  Following the technique from [3] a model M, which is a $m \times m$ matrix, is constructed using W and H. Let $\vec{x}$ be a binary ingredients vector (akin to that detailed in IP2), then:

  $$\vec{y} = \vec{x}M \tag{4}$$

  The product $\vec{y}$ is a numerical vector of size m, where greater values denote closer relation to the ingredients present in $\vec{x}$. As with the previous predictor, the top-k predictions correlate to ingredients with the highest values in the resulting vector, $\vec{y}$.

2) Amounts predictor: This component is tasked with predicting the gram-amounts a set of ingredients is likely to be used in. 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 (for example a kilo of onions and 10 grams of flour) to gain intuition about real-world ratios we've created AN. The construction of the network is similar to the network used for IP2. The difference between IP2 and the amounts predictor lies in the usage of the network's output. While the ingredient predictor checks the values of excluded ingredients, here only the present ingredients are of interest. For each of the contained ingredients we extract from the output vector the matching values, these pose as an approximation of the probable amounts.

3) Model Training: To train the different models it is necessary to represent the recipes' data accordingly.

- IP1 is the only model to involve no training. It requires merely the construction of the analogous graph.
- IP2 and the amount predictor are trained using the Keras framework[1]. 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.
- IP3 also requires the binary recipe-ingredient matrix X. Training this model consists of decomposing X into matrices W and H, which are used to establish the model M.

4) Best ingredient choice: 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:

$$MSE = \frac{1}{|N|} \sum_{i \in N} (t_i - p_i)^2 \tag{5}$$

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 Table I). Additionally, we did 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.

B. Step 2: Recipe selection

We now use the pseudo-recipe constructed in step 1 in order to search the dataset for the top-k most similar recipes (*FindSimilarRecipes*, Algortihm 1, line 7). 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 don't necessarily coincide. To mitigate this issue, we prescribe *similarity* as a weighted average of the Jaccard and cosine metrics:

$$\text{sim} = \text{COS}_{\text{weight}} \cdot \text{sim}_{\text{cos}} + (1 - \text{COS}_{\text{weight}}) \cdot \text{sim}_{\text{jaccard}} \tag{6}$$

[1] F. Chollet, 2015. Keras. https://keras.io

Cosine similarity captures the likeness in terms of amounts:

$$\mathrm{sim}_{\cos} = \frac{a \cdot b}{|a| \cdot |b|} = \frac{\sum_{i=1}^{m} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{m} a_i^2}\sqrt{\sum_{i=1}^{m} b_i^2}}, \tag{7}$$

where a is the pseudo-recipe amounts vector and 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:

$$\mathrm{sim}_{\mathrm{jaccard}} = \frac{|A \cap B|}{|A \cup B|}. \tag{8}$$

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

IV. Datasets

The datasets utilized in this paper were collected from Allrecipes[2] and Yummly[3]. All nutritional values for the recipes are available on the websites. The ingredients and amounts were parsed in a semiautomatic manner, using fuzzy matching between the recipes' ingredients and the foods in the USDA dataset[4]. About 80% of ingredients were recognized and at least 70% were parsed. A parsed ingredient is a recognized ingredient which we were able to quantify in grams. Table I displays the datasets characteristics. The WHO scores were computed using the manner described in [2], their distribution is displayed in Fig. 2. 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.

TABLE I
DATASETS STATISTICS

	Allrecipes	**Yummly**
Number of recipes	37032	90798
Average no. of ingredients per recipe	11	13
Recognized ingredients	82%	81%
Parsed ingredients	77%	71%
Unique ingredients	465	512

V. Evaluation and Results

A. Evaluation on proposed models

We attentively design the experiments and evaluate the performance of our methods from different aspects. First, all models presented will be assessed separately. Then, we'll define the evaluation method of the complete framework and examine the experiments' results.

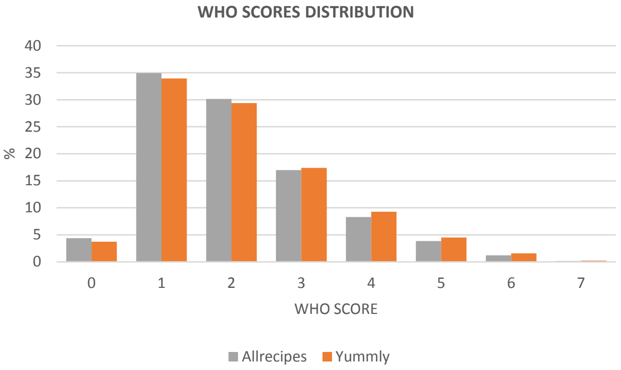


Fig. 2. WHO scores distribution in Allrecipes and Yummly datasets. WHO score reflects how well a recipe meets the WHO guideline on the scale of 0-7 (poorly-well). A recipe with a high WHO score is considered healthy.

1) Ingredients predictor models: The ingredient predictors are 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 predictor IP3 in [3]. The results of this evaluation are reported as:

- Percent of predictions with the removed ingredient ranked ≤ 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'd like the mean and median ranks to be low and the percent of good predictions to be as high as possible.

This experiment is repeated five times for all predictors. IP3 is tested with 2-4 components (i.e. number of hidden factors). The results of this evaluation are summarized in Table II. It is noted that the results for IP3 are quite similar to the results reported in [3].

IP1 produces the best results out of the three models. It is able to include the correct ingredient within the top-10 elements in 53% of the cases on the Allrecipes dataset and 51% of the cases on the Yummly dataset. Its mean and median scores are also lower than the competitors. From our results we conclude the best c value for IP3 is 2. Meaning two latent factors are sufficient to represent relations between ingredients. This parameter is used in all subsequent experiments.

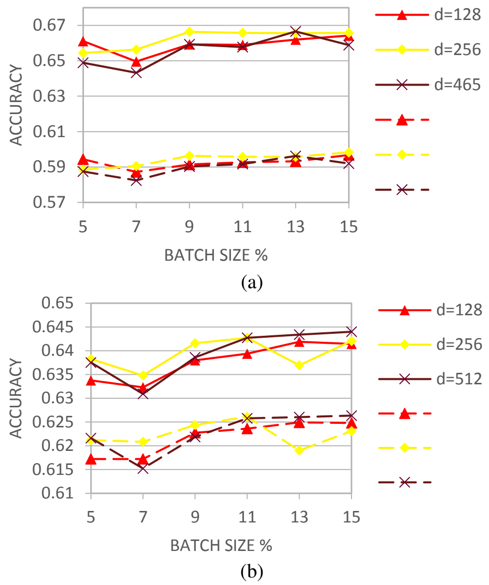
TABLE II
EVALUATION OF INGREDIENT PREDICTORS MODELS

		IP1	**IP2**	**IP3**		
				c=2	***c=3***	***c=4***
Allrecipes	Rank $\leq$ 10 Mean rank Median rank	**53%** **22.22** **9**	28% 52.54 26	50% 37.09 10	49% 41.00 11	48% 43.17 12
Yummly	Rank $\leq$ 10 Mean rank Median rank	**51%** **26.09** **10**	21% 56.56 34	50% 37.80 11	49% 43.40 11	49% 47.01 11

[2] https://www.allrecipes.com

[3] https://www.yummly.com

[4] 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



(a)

(b)

Fig. 3. Amounts predictor accuracy results for Allrecipes (a) and Yummly (b). Solid lines represent training-set accuracy and dashed lines represent the accuracy on the validation set.

2) *Amounts predictor model:* The amounts predictor is evaluated by the AN accuracy. The evaluation is performed on a 70-30 validation split while varying the hidden layer and batch sizes. A batch size of 15% produces the best results on both datasets. The highest train and validation accuracy scores for Allrecipes are 66.56% and 59.82% respectively with a hidden layer dimension of 256. For Yummly the highest accuracy scores obtained are 64.39% for the train set and 62.63% on the validation set with a hidden layer the size of the input layer, 512 neurons. Fig. 3 illustrates these results. It is not surprising a higher validation accuracy is observed on Yummly as its size provides more training opportunities. The overall accuracy of this model is not as high as that of novel neural-network applications, however recall this predictor is merely a component within a larger framework. Our objective is to demonstrate the effectiveness of the general algorithm and reasonably a more robust predictor will only be favorable.

3) *Complete framework:* The evaluation of the algorithm is performed using 120 most frequent item sets. Frequent sets are sets of ingredients which repeatedly appear together in recipes. The algorithm parameters are set to n=5 and k=10. Meaning, the algorithm increases the initial set by 5 ingredients and returns the top-10 most similar recipes.

The performance is reported using the mean WHO score measure. Mean WHO score is based on the recommendations' WHO score distribution and is compared to 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 mean WHO score, we have studied another error-based metric, using logic similar to Eq. 5, in our preliminary experiments. However, our tests proved this measure to be inappropriate for

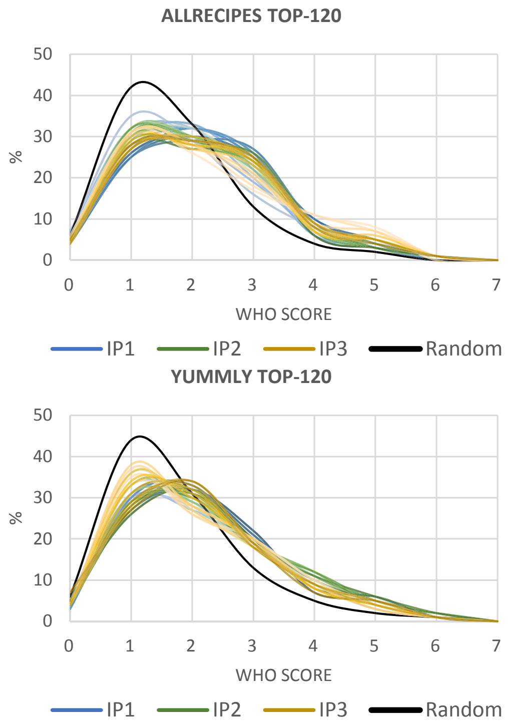


Fig. 4. WHO scores distribution of recommended recipes using different ingredient predictors and COS values. Color codes: IP1=Blue, IP2=Green, IP3=Yellow, the black outline denotes the random baseline. Brighter colors correspond to lower COS values. The percentages for the ingredient predictors are computed with respect to top-10 recommendations.

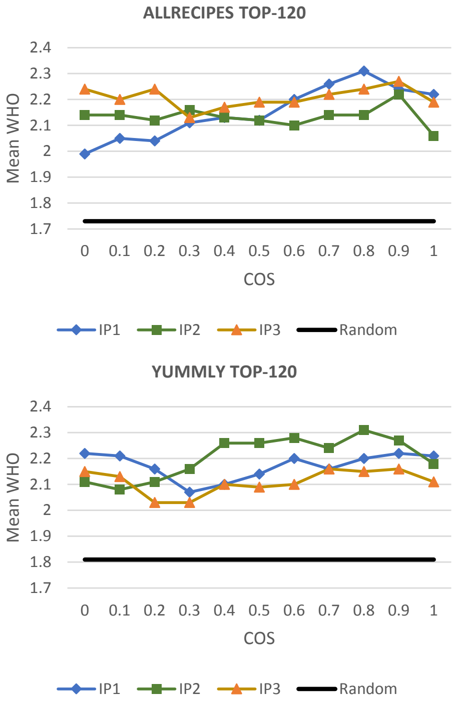


Fig. 5. Mean WHO score of recommended recipes using different ingredient predictors.

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 Table III. 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.

TABLE III
TARGET MACRO-NUTRIENTS

Nutrient	WHO guideline	Target[a]
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

[a]Percentage conversion is based on 2000 Kcal/day diet.

The WHO scores distribution of the recommendations is demonstrated in Fig. 4. The colors correspond to the three ingredients predictors, IP1 is denoted by blue lines, IP2 is green and IP3 is yellow. The different shades correlate to the COS$_{\text{weight}}$ (COS) in Eq. 6, brighter lines mean lower cosine-similarity weight (i.e. higher Jaccard-similarity weight). The random-draw distribution is laid out by the black outline. Observing the distribution charts, we can see both datasets display similar patterns. The brighter lines resemble the baseline plot more than the darker ones. The peaks of the bright-distributions are lower than the baseline's mode but their values are quite similar, about 1.2. The dark-distributions' maximums are found closer to 2. The separation between the lines is somewhat clearer for Yummly due to its size, having more recipes to choose from results in less overlap in the recommendations. Even though, all variations preserve the right-skewed pattern of the original dataset, we notice they all surpass the baseline in the 3 to 5 WHO score range. This means more recipes with higher scores are recommended. Recipes with WHO scores of 6 and 7 are extremely rare in both datasets, as witnessed from Fig. 2, hence there is no visible improvement in their retrieval.

Overall, it is evident that all ingredient predictors, regardless of the COS set, are able to nudge the distribution towards the desired end. To examine this result further, the mean WHO score is considered. Fig. 5 shows the change in the mean WHO score of the predictions as a function of COS. This chart further illustrates that all COS settings improved the baseline. Additional investigation reveals that higher cosine-similarity weights actually preformed slightly better. The baseline mean WHO score for the top-120 most frequent sets in Allrecipes and Yummly are 1.73 and 1.81 respectively. The proposed algorithm is able to improve the mean WHO score on both datasets to 2.31, using COS=0.8 with IP1 for Allrecipes and COS=0.8 with IP2 on Yummly. We note that for both datasets employing just ingredient similarity or quantity similarity (i.e. COS=0 or COS=1) did not bring about the best performance.

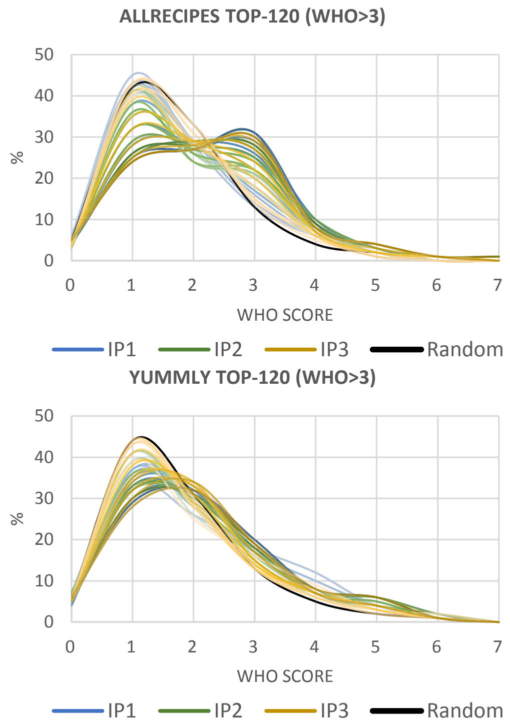


Fig. 6. WHO scores distribution of recommended recipes, using different ingredient predictors, when training the amounts network on recipes with WHO $>$ 3. Color codes are similar to Fig.4.

B. *Comparison with alternative methods*

As no state-of-the-art method for comparison is available, we formulate an alternative design and validate our proposal through empirical analysis. Inspired by our prior results, we explore whether the mean WHO score can be further increased by training the AN using only recipes with WHO scores greater than 3 (i.e. using a dataset with mostly healthy examples).

The results are illustrated in Fig. 6 and Fig. 7. The distribution chart for Yummly does not display any major change, yet the Allrecipes distribution seems to hold some promise. However, the actual improvement in the mean WHO score is negligible. The highest mean WHO attained for Allrecipes is 2.33 (IP1 and IP3, COS=1), a minor 0.02 gain. For Yummly the top score is 2.21 (IP1, COS=1) which is in fact a decrease of 0.1.

We will point out two striking trends apparent in the results. First, note that in Fig. 6 the brighter lines appear in larger proximity to the baseline plot than in Fig. 4. Now, recall the ingredient selection at each step of the algorithm relies on

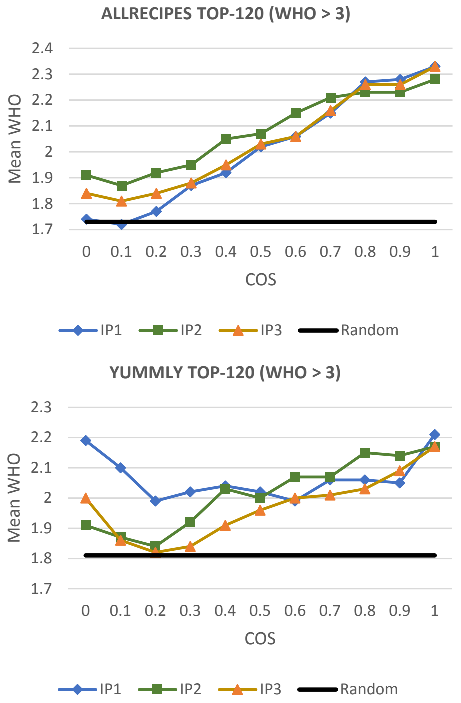


Fig. 7. Mean WHO score of recommended recipes, using different ingredient predictor, when training the amounts network on recipes with WHO > 3.

the ability to predict a set's quantities. As can be deduced from Fig. 2, 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. 7 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. 7, is the positive correlation of COS and mean WHO score, unlike the vague relation seen in Fig.5. 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.

VI. Future work and conclusion

The method presented in this paper is an attempt to utilize machine-learning techniques to tackle the healthy-recipe recommendation problem. 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 three 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, for instance low-fat or high-protein diets. Finally, as 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.

Acknowledgment

This work was supported in part by the NSERC Discovery Grants, and the National Natural Science Foundation of China under Grant No. 61572289.

References

[1] S. Sajadmanesh, S. Jafarzadeh, S. A. Ossia, H. R. Rabiee, H. Haddadi, Y. Mejova, M. Musolesi, E. D. Cristofaro, and G. Stringhini, “Kissing cuisines: Exploring worldwide culinary habits on the web,” in *Proc. 26th Int. Conf. World Wide Web Companion*, ser. WWW ’17 Companion, 2017, pp. 1013–1021.

[2] C. Trattner and D. Elsweiler, “Investigating the healthiness of internet-sourced recipes: Implications for meal planning and recommender systems,” in *Proc. 26th Int. Conf. World Wide Web*, ser. WWW ’17, 2017, pp. 489–498.

[3] M. D. Clercq, M. Stock, B. D. Baets, and W. Waegeman, “Data-driven recipe completion using machine learning methods,” *Trends in Food Science & Technology*, vol. 49, pp. 1 – 13, 2016.

[4] J. Freyne and S. Berkovsky, “Intelligent food planning: Personalized recipe recommendation,” in *Proc. 15th Int. Conf. Intelligent User Interfaces*, ser. IUI ’10, 2010, pp. 321–324.

[5] C.-Y. Teng, Y.-R. Lin, and L. A. Adamic, “Recipe recommendation using ingredient networks,” in *Proc. 4th Ann. ACM Web Science Conference*, ser. WebSci ’12, 2012, pp. 298–307.

[6] Allrecipes. (2018) Allrecipes — food, friends, and recipe inspiration. [Online]. Available: https://www.allrecipes.com

[7] Yummly. (2018) Yummly: Personalized recipe recommendations and search. [Online]. Available: https://www.yummly.com/

[8] D. Elsweiler, C. Trattner, and M. Harvey, “Exploiting food choice biases for healthier recipe recommendation,” in *Proc. 40th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, ser. SIGIR ’17, 2017, pp. 575–584.

[9] S. Howard, J. Adams, and M. White, “Nutritional content of supermarket ready meals and recipes by television chefs in the united kingdom: cross sectional study,” *Bmj*, vol. 345, p. e7607, 2012.

[10] D. Elsweiler and M. Harvey, “Towards automatic meal plan recommendations for balanced nutrition,” in *Proc. 9th ACM Conf. Recommender Systems*, ser. RecSys ’15, 2015, pp. 313–316.

[11] D. H. Fudholi, N. Maneerat, and R. Varakulsiripunth, “Ontology-based daily menu assistance system,” in *2009 6th Int. Conf. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, vol. 02, May 2009, pp. 694–697.

[12] V. Espín, M. V. Hurtado, and M. Noguera, “Nutrition for elder care: a nutritional semantic recommender system for the elderly,” *Expert Systems*, vol. 33, no. 2, pp. 201–210, 2016.

[13] P. Tumnark, F. A. d. Conceição, J. P. Vilas-Boas, L. Oliveira, P. Cardoso, J. Cabral, and N. Santibutr, “Ontology-based personalized dietary recommendation for weightlifting,” in *Proc. 2013 Int. Workshop Computer Science in Sports*. Atlantis Press, 2013, pp. 44–49.

[14] N. Suksom, M. Buranarach, Y. M. Thein, T. Supnithi, and P. Netisopakul, “A knowledge-based framework for development of personalized food recommender system,” in *Proc. 5th Int. Conf. Knowledge, Information and Creativity Support Systems*, 2010.

[15] Y. Kikuchi, M. Kumano, and M. Kimura, “Analyzing dynamical activities of co-occurrence patterns for cooking ingredients,” in *2017 IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov 2017, pp. 17–24.

[16] S. Yokoi, K. Doman, T. Hirayama, I. Ide, D. Deguchi, and H. Murase, “Typicality analysis of the combination of ingredients in a cooking recipe for assisting the arrangement of ingredients,” in *2015 IEEE Int. Conf. Multimedia & Expo Workshops (ICMEW)*, June 2015, pp. 1–6.