Visually-aware Collaborative Food Recommendation

Xiaoyan Gao¹, Fuli Feng², Xiangnan He², Heyan Huang¹, Xinyu Guan³, Chong Feng¹, Zhaoyan Ming², and Tat-Seng Chua²
¹Department of Computer Science, Beijing Institute of Technology
²School of Computing, National University of Singapore
³Xi’an Jiaotong University

Abstract

Food recommender systems play an important role in assisting users to identify the desired food to eat. Deciding what food to eat is a complex and multi-faceted process, which is influenced by many factors such as the ingredients, appearance of the recipe, user’s personal preference on food, and various contexts like what had been eaten in past meals. In this work, we formulate the food recommendation problem as predicting user preference on recipes based on three key factors that determine a user’s choice on food, namely, 1) the user’s (and other users’) history; 2) the ingredients of a recipe; and 3) the descriptive image of a recipe. To address this challenging problem, we develop a dedicated neural network-based solution Hierarchical Attention based Food Recommendation (HAFR) which is capable of: 1) capturing the collaborative filtering effect like what similar users tend to eat; 2) inferring a user’s preference at the ingredient level; and 3) learning user preference from the recipe’s visual image. To evaluate our proposed method, we construct a large-scale dataset consisting of millions of ratings from AllRecipes.com. Extensive experiments show that our method outperforms several competing recommender solutions like Factorization Machine and Visual Bayesian Personalized Ranking with an average improvement of 12%, offering promising results in predicting user preference on food.

1 Introduction

The advent of Internet and mobile technologies facilitates people to access information at any time and place. People’s lifestyles have been changed profoundly with the spread of online services, such as social media, E-commerce and various lifestyle Apps. For example, when friends have dinner together, one may take pictures of food she likes and shares the pictures in social media; when deciding what to eat, users may resort to food-related Apps for recommendations.

“Food is the first necessity of the people”. Because of the importance of food in everyday life, food recommender systems have become an indispensable component in many lifestyle services, and are often touted as potential means to affect people towards a healthy lifestyle (Trattner and Elsweiler 2017). To build a successful food recommender system, the first step is to accurately understand a user’s food preference (Harvey, Ludwig, and Elsweiler 2012). Even for building health-driven food services (Ge, Ricci, and Masimo 2015), only when the recommended food (or recipe) meet the taste of a user, the user can be persuaded to follow the recommendation.

In this work, we focus on the fundamental problem in building food recommender system — inferring user preference on food. While many efforts have been devoted to recommendation techniques (He et al. 2017; He et al. 2018), they largely focused on the general domains of Movies or E-Commerce products, and relatively little attention has been paid to the specific domain of food. We argue that food is a special class of problem that requires dedicated methods to predict user preference on food:

- Food is not atomic — a recipe usually consists of multiple ingredients. In most cases, people prefer a recipe or dish, just because it contains the ingredients they like to eat. An example is shown in Figure 1, where we sample two users and show their recent four food choices. We find that the first user (male, first row) likes spicy food, as evidenced by the frequently appearing chili in his chosen food. In contrast, the second user (female, second row) likes seafood, as evidenced by the shrimp in her food choices. As such, to accurately learn user preference on food, it is insufficient to represent a recipe as an ID and simply run collaborative filtering (CF) algorithms like matrix factorization (He et al. 2016b).

- Food image conveys rich information beyond ingredients, being crucial in affecting a user’s preference on the food. For example, different cutting and cooking methods can

---

¹sumitted to AAAI2019

2We interchangeably use the term “food” and “recipe” which refer the same thing in this paper.
make the same ingredient taste quite differently. “A picture is worth a thousand words” — it is more intuitive (and often easier) for a user to determine the taste of a recipe from its image than from its textual description. As such, it is crucial to properly account for the visual semantics in food image to provide quality recommendation service.

To the best of our knowledge, none of the existing work has approached food recommendation with a comprehensive consideration of all above-mentioned factors. Existing work has either adopted collaborative filtering which is limited to user-food interaction modeling (Harvey, Ludwig, and Elsweiler 2012), or performed content-based filtering based on the ingredients (Freyne and Berkovsky 2010a) or image features (Yang et al. 2015). We argue that both user-food interactions and the content features (i.e., ingredients and food image) are highly important in food recommendation, such that they should be carefully coupled to infer users’ preference on food. Specifically, different users may consume the same food because of different ingredients, and the impact of content features and collaborative effects could also vary for different users. These properties need to be explicitly captured and jointly modeled in the method.

Targeting at developing dedicated recommendation solution for the food domain, we make the following contributions in this paper:

• First, we present a new problem formulation, aiming to comprehensively account for the key factors that affect a user’s food decision-making process, including user-food interaction history, food ingredients, and food image.

• Second, we propose a new neural network solution named HAFR, leveraging the strong representation capability of multi-layer neural network and the interpretability of attention mechanism to jointly model the three key factors.

• Lastly, we contribute a new dataset for evaluating this task, which is constructed from AllRecipes.com containing over a million ratings. We evaluate our HAFR method on this dataset and demonstrate superior performance over competing recommendation methods.

In the remainder of this paper, we first present the problem formulation in Section 2 followed by elaborating the method in Section 3 and experimental results in Section 4. Lastly, we discuss related work in Section 5 and conclude the paper, and envision some future work in Section 6.

2 Problem Formulation

The problem setting of food recommendation is to predict a user’s preference on recipes from: historical user-recipe interactions; recipe image; and recipe ingredients. Given a set of users (U) and recipes (I), we use a binary matrix \( Y \in \mathbb{R}^{M \times N} \) to denote the user-recipe interactions, where \( M \) and \( N \) denote the number of users and recipes, respectively. Each entry \( y_{ui} \) denotes whether a user \( u \) has interacted with \( i.e., \) rated \( i \), which is defined as:

\[
y_{ui} = \begin{cases} 1, & \text{if user } u \text{ consumed recipe } i; \\ 0, & \text{otherwise.} \end{cases}
\]

\[ (1) \]

Besides the ID \( i \), a recipe also contains an image \( V_i \) and a set of ingredients \( g_i \). \( V_i \) denotes the raw image feature, \( g_i \in \mathbb{R}^K \) is a multi-hot encoding with \( g_i^t = 1 \) denoting that ingredient \( t \) is in recipe \( i \), where \( K \) is the number of ingredients occurred in \( I \), \( t \) is the ID of ingredient. Then, our task is to learn an interaction function \( \hat{y}_{ui} = f(u, i, g_i, V_i) \) which estimates the probability that user \( u \) would interact with recipe \( i \), which is formally defined as:

\[ \text{Input: Users } U, \text{ recipes } I, \text{ user-recipe interactions } Y, \text{ recipe ingredients } \{g_1, \ldots, g_N\}, \text{ and recipe images } \{V_1, \ldots, V_N\}. \]

\[ \text{Output: An interaction function } \hat{y}_{ui} = f(u, i, g_i, V_i), \] which outputs the probability that user \( u \) would consume recipe \( i \).

3 Methodology

In this section, we introduce the proposed HAFR for solving the food recommendation problem. Different from existing food recommender systems, which either mine the user-recipe interactions or analyze recipe contents (Trattner and Elsweiler 2017; Yang et al. 2015), we explicitly incorporate ingredient and visual image as key information to infer users’ preference over recipes. We present the architecture of HAFR model in Figure 2, which consists of three main components:

• **Embedding layer** encodes user-recipe interaction (user ID and recipe ID), recipe image, and recipe ingredients as the embedding of user, recipe, image, and ingredients, respectively.

• **Hierarchical attention** dynamically aggregates the embedding into a more comprehensive recipe representation (\( q_S \)) by accounting for the preference of the target user.

• **Output layer**, in which we compose the vector representation of the user and the recipe, which is fed into a multi-layer perceptron (MLP) to estimate \( \hat{y}_{ui} \).

The remainder of this section is organized to elaborate the three components one by one.

3.1 Embedding Layer

**User-recipe interaction.** Embedding-based models (He et al. 2017; Wang et al. 2018) associate each user and item with
an embedding (i.e., a real-valued vector), which has become the mainstream to model user-item interactions. Considering their recent success, we also project the user and recipe IDs (i.e., u and i) into the embeddings, which are expected to encode the collaborative signals among users and recipes. Formally, we first encode u (i) into an one-hot encoding \( e_u \in \mathbb{R}^M \) \( (e_i \in \mathbb{R}^N) \) with zero values except the u-th (i-th) entry. We then employ two embedding layers to project the one-hot encodings into embeddings, which are formulated as,

\[
p_u = Pe_u, \quad q_i = Qe_i,
\]

where \( p_u \) and \( q_i \) are parameters to be learned. Note that we only use one mapping layer for the projection instead of stacking multiple layers for the consideration of model complexity.

**Recipe Image.** As a significant information carrier, recipe image presents rich and intuitive information and plays an important role in affecting users’ preference and selection on recipes (Elswieker, Trattner, and Harvey 2017; Chokr and Elbassuoni 2017). As such, we incorporate the image of a recipe to enhance its representation. Inspired by its success in many computer vision tasks, we employ the ResNet-50 (He et al. 2016) to extract visual features \( v_i \) from the raw image \( V_i \). Note that \( v_i \) is a 2,048 dimension vector from the pool5 layer of ResNet-50. Besides, the ResNet-50 is pre-trained on ImageNet (Deng et al. 2009) and fine-tuned via classifying raw recipe images into their associated categories (e.g., chicken rice).

In order to learn all representations in the same latent space, we then project \( v_i \) into the dimension of \( D \) through a mapping layer, of which the formulation is,

\[
m_i = Wv_i + b,
\]

where \( m_i \in \mathbb{R}^D \) is termed as image embedding, \( W \) and \( b \) are parameters to be learned. Note that we only use one mapping layer for the projection instead of stacking multiple layers for the consideration of model complexity.

**Recipe Ingredients.** Previous work adopting content-based filtering for food recommendation has shown that ingredients of a recipe could be important clues for inferring user preference over the recipe (Freyne and Berkovsky 2010a). For instance, a user would consume a recipe simply because it contains the favourite ingredient of the user. As such, we further encode the ingredients of a recipe to enrich its representation. Note that the multi-hot encoding \( q_i \in \mathbb{R}^K \) is a sparse (\( K \) is typically larger than ten thousand) vector with binary values. To overcome the sparsity problem, we also learn an embedding for each ingredient. Formally, we denote the ingredient embeddings as \( X = [x^1, \ldots, x^K] \in \mathbb{R}^{D \times K} \), where \( x^k \) is the embedding of ingredient \( k \). From \( X \), we then obtain embedding of ingredients occurred in recipe \( i \), i.e.,

\[
\mathcal{X}_i = \{x^k | g_{ik} = 1\}, \quad \text{where} \quad x^k = l(u(k, X), \quad (4)
\]

recall that \( g_i \) is the multi-hot encoding of the ingredient list.

### 3.2 Hierarchical Attention

Motivated by the inherent hierarchical character of the recipe information, \( i.e., \) a recipe consists of three components interaction signals \((q_i)\), image features \((m_i)\), ingredients \((X_i)\) that can contains multiple ingredients, we devise a hierarchical attention to aggregate recipe information from ingredient-level to component-level.

**Ingredient-level Attention.** Considering that different recipes would contain different numbers of ingredients, we employ a pooling operation to aggregate the ingredient embeddings into a single one to represent an ingredient list, so as to avoid the problem of variant length. Our first intuition is equally fusing the ingredient embeddings with an average pooling,

\[
\bar{x}_i = \frac{1}{|\mathcal{X}_i|} \sum_{g_{ik} = 1} x^k_i,
\]

\(|\mathcal{X}_i|\) denotes the size of \( \mathcal{X}_i \). However, the average pooling lacks the flexibility to dynamically adjust the weights of ingredients, which is particularly useful when a user is more interested in certain ingredients. For instance, a user enjoys chilli might have higher probability to select recipes containing chili. As such, the embedding of chili may contribute more than other ingredients to the compressed embedding regarding the given user. Therefore, we argue that the embedding of ingredient should be fused with adaptive weights to capture users’ dynamic preference over ingredients, of which the formulation is,

\[
\bar{x}_i = \sum_{g_{ik} = 1} \alpha^k(u)x^k_i.
\]

Inspired by the recent success of attention mechanism (Bahdanau, Cho, and Bengio 2015; Xiao et al. 2017), which allows different parts to contribute differently when compressing them to a single representation, we propose an ingredient-level attention to model the coefficients \( \alpha^k(u) \). Specifically, as illustrated in Figure 3 we use a two-layer network to compute \( \alpha^k(u) \) with user embedding \( p_u \), image
embedding \( m_i \), and the embedding of the target ingredient \( x^k_i \) as input,
\[
\alpha^k_i(u) = \exp(a^k_i(u)) / \sum_{q^k_i=1} \exp(a^k_i(u)), \tag{7}
\]
\[
a^k_i(u) = h^T \tanh(W_{1p}p_u + W_{1m}m_i + W_{1x}x^k_i + b_1),
\]
where \( W_{1s}, b_1, \) and \( h \) are the parameters to be learned. \( \tanh \) is hyperbolic tangent function.

**Component-level Attention.** Currently, we have three representations \((q_i, m_i, x_i)\) with the same dimension, corresponding different components of the recipe. One naive way to jointly account for the components, i.e., aggregating the interaction signals and content features, is separately estimating three probabilities from \((p_u, q_i), (p_u, m_i),\) and \((p_u, x_i)\), and merging them into the overall probability of interaction between \( u \) and \( i \). However, this would present issues due to the increase of model complexity. Therefore, we first aggregate \( q_i, m_i, \) and \( x_i \) into a more comprehensive food representation, encoding both user-recipe interactions and the content-features.

Similar as fusing ingredient embeddings into \( x_i \), we aggregate the three components \((i.e., q_i, m_i, \) and \( x_i)\) via,
\[
\tilde{q}_i = \sum_{c_p \in \{q_i, m_i, x_i\}} \beta_p(u)c_p, \tag{8}
\]
where \( \beta_p(u) \) is a user-aware coefficient, adjusting the importance of different components. Note that we use \( c_p \) to represent the component representation to simplify the illustration. The motivation of incorporating the user-aware coefficients is to capture users’ dynamic preference over different components. For instance, one user might choose food for their attracting appearance, whereas another user would prefer food similar to their accustomed tastes. Similar as the ingredient-level attention, we devise a component-level attention module to compute the user-aware coefficients. Specifically, the component-level attention module is also a two-layer network formulated as,
\[
\beta_p(u) = \frac{\exp(b_p(u))}{\sum_{c_p \in \{q_i, m_i, x_i\}} \exp(b_p(u))},
\]
\[
b_p(u) = v^T \tanh(W_{2p}p_u + W_{2c}c_p + b_2), \tag{9}
\]
where the matrices \( W_{2s} \) and bias \( b_2 \) are parameters to be learned.

It should be noted that hierarchical attention network (HAN) is originally proposed to learn document representations, where a word-level attention extracts important words for sentence representation and a sentence-level attention rewards sentences providing clues for downstream applications (Yang et al. 2016). Despite following the original idea of HAN, our proposed hierarchical attention is different from HAN for two reasons: 1) besides the contents of the recipe, both of the ingredient-level and component-level attentions calculate coefficients conditioned on user embedding, capturing users’ dynamic preference; 2) our hierarchical attention aggregates heterogeneous inputs \((i.e., user-recipe interactions, image, and ingredients)\) rather than homogeneous ones \((i.e., words)\).

### 3.3 Output Layer and Training

**Output Layer** Until now, given a pair of user \((u)\) and recipe \( i \) with the associated image and ingredients, we obtain the embeddings for user \( p_u \) and recipe \( q_i \). Following [He et al. 2017], we compose a representation for the user-recipe pair by concatenating user embedding, recipe embedding, and their element-wise product \((p_u \odot q_i)\). Inspired by the recent deep recommendation methods (Hu et al. 2018), above the concatenation layer, we feed the unified representation into an MLP in order to deploy the nonlinear function for modeling complicated interactions. Formally, the output layer estimates the interaction probability between user and recipe as:
\[
\hat{y}_{ui} = z^T f(W_3 \begin{bmatrix} p_u \\ \tilde{q}_i \\ p_u \odot \tilde{q}_i \end{bmatrix} + b_3) \tag{10}
\]
where, \( W_3, b_3, \) and \( z \) denote the weight matrix, bias vector, and weights of output layer, respectively. \( f(\cdot) \) is set to the ReLU function, which has empirically shown to work well.

**Objective Function** Since we address the food recommendation task from the ranking perspective, we employ a pairwise learning method to optimize model parameters. The assumption of pairwise learning is that the model could predict a higher score for an observed interaction than unobserved counterparts. Specifically, an observed user-recipe interaction is assigned to a target value \( 1 \), otherwise \( 0 \). We adopt Bayesian Personalized Ranking (BPR) loss, which is widely used in recommendation (Chen et al. 2017):
\[
\mathcal{L} = \sum_{(u,i,k) \in \mathcal{D}_S} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uk}) + \lambda ||\Theta||^2, \tag{11}
\]
where \( \sigma \) is the logistic (sigmoid) function and \( \lambda \) is a model specific regularization hyperparameter. \( \Theta \) is the parameters in our model. \( \mathcal{D}_S \) is the training set, it consists of triples in the form \((u, i, k)\), where \( u \) denotes the user together with an interacted recipe \( i \) and a non-observed recipe \( k \):
\[
\mathcal{D}_S = \{(u, i, k) | y_{ui} = 1 \land y_{uk} = 0\} \tag{12}
\]
To optimize the above objective function, we adopt a mini-batch Adagrad (Duchi, Hazan, and Singer 2011), a variant of Stochastic Gradient Descent (SGD) that applies an adaptive learning rate for each parameter. In each training epoch, we sample the negative instance \((i.e., sampling k for (u, i))\) on-the-fly, as done in Bayesian Personalized Ranking (Rendle et al. 2009).

### 4 Experiments

We conducted extensive experiments on a real-world dataset to answer the following research questions:

- **RQ1:** How does our proposed model perform as compared with the state-of-the-art models that are designed for food recommendation?
- **RQ2:** Is the proposed hierarchical attention helpful for learning recipe representation and improving the recommendation performance?
In what follows, we first describe the experimental settings, followed by answering the above research questions.

4.1 Experimental Settings

Data Collection. Because of the lack of public datasets with rich user-recipe interactions, we construct a dataset via collecting data from Allrecipes.com. The website is chosen since it is one of the largest food-oriented social networks with 1.5 billion visits per year. In particular, we crawl 52,821 recipes from 27 categories posted between 2000 and 2018. For each recipe, we crawl its ingredients, image and the corresponding ratings from users. The ratings are transformed into binary implicit feedback as ground truth, indicating whether the user has interacted with the specific recipe. This is because we focus on modeling users’ preference over recipes, where the rating activity reflects a strong interest on the recipe (the user has tried the recipe). To ensure the quality of the data, we filter recipes that do not have images.

To evaluate model performance, we holddout the latest 30% of interaction history to construct the test set, and split the remaining data into training (60%) and validation (10%) sets. Then we retain users which occur in both training and test sets, and obtain 68,768 users, 45,630 recipes with 33,147 ingredients and 1,093,845 interactions. Moreover, we depict the distribution of user activity and recipe popularity on the constructed dataset in Figure 4. We can see that many recipes (users) only have few ratings, i.e., most recipes (users) lie in the long tail of the distribution. Note that we will release the dataset once the paper is accepted.

Evaluation Protocol. Given one user and its rated recipes in the test set, we sample 500 negative instances that the user has not interacted. Considering the long tail distribution of recipe popularity, we bias the negative sampling towards the recipes that we will release the dataset once the paper is accepted.

4.1 Experimental Settings

Data Collection. Because of the lack of public datasets with rich user-recipe interactions, we construct a dataset via collecting data from Allrecipes.com. The website is chosen since it is one of the largest food-oriented social networks with 1.5 billion visits per year. In particular, we crawl 52,821 recipes from 27 categories posted between 2000 and 2018. For each recipe, we crawl its ingredients, image and the corresponding ratings from users. The ratings are transformed into binary implicit feedback as ground truth, indicating whether the user has interacted with the specific recipe. This is because we focus on modeling users’ preference over recipes, where the rating activity reflects a strong interest on the recipe (the user has tried the recipe). To ensure the quality of the data, we filter recipes that do not have images.

To evaluate model performance, we holdout the latest 30% of interaction history to construct the test set, and split the remaining data into training (60%) and validation (10%) sets. Then we retain users which occur in both training and test sets, and obtain 68,768 users, 45,630 recipes with 33,147 ingredients and 1,093,845 interactions. Moreover, we depict the distribution of user activity and recipe popularity on the constructed dataset in Figure 4. We can see that many recipes (users) only have few ratings, i.e., most recipes (users) lie in the long tail of the distribution. Note that we will release the dataset once the paper is accepted.

Evaluation Protocol. Given one user and its rated recipes in the test set, we sample 500 negative instances that the user has not interacted. Considering the long tail distribution of recipe popularity, we bias the negative sampling towards the popular recipes. Specifically, we calculate the popularity of a recipe \( f_i \) on the training set, and sample it with a probability of \( \frac{f_i}{\sum_{i' \in I} f_{i'}} \). \( r \) is a coefficient to control the bias (larger \( r \) leads to higher probability for popular recipes), which is empirically set as \( r = 0.7 \). Note that this sampling strategy is unfriendly to methods with popularity bias (i.e., potentially predicting popular recipes as positive), and benefit methods that focus on inferring users’ preference, which is the key target of food recommendation.

We then estimate the interaction probability of each instance and generate a ranking list. To evaluate the performance, we adopt three popular metrics for recommendation, Area Under the Roc Curve (AUC), Normalized Discounted Cumulative Gain (NDCG) and Recall (Rendle et al. 2009). AUC measures the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. NDCG@k assigns higher scores to the hits at higher positions of the ranking list. Recall@k is the probability that positive items are ranked in the top-k positions. For all metrics, the values range from 0 to 1 (the higher the better). Note that we report the average score for 10 different repeats of testing, and apply t-test to obtain the statistical significance.

Baselines. To justify the effectiveness of our proposed model, we compared the performance of our proposed model with the following baselines. Note that all methods are learned by optimizing the same pairwise ranking loss of BPR for fair comparison. The compared recommendation methods are introduced as follows:

- LDA (Trattner and Elsweiler 2017) is the state-of-the-art food recommendation method. Following (Trattner and Elsweiler 2017), we use the Librec implementation and tune the topic number.
- Matrix Factorization-Bayesian Personalized Ranking (MF-BPR) (Rendle et al. 2009) optimizes the standard MF model with the pairwise BPR ranking loss. This method is a popular choice for building a CF recommender from implicit feedback.
- Factorization Machine (FM) (Rendle et al. 2011) estimates the target by modeling all interactions between each pair of features via factorized interaction parameters. In this work, we set input features as user ID, recipe ID, and ingredients.
- VBPR (He and McAuley 2016) adds visual contents to MF-BPR, using the visual contents as parts of recipe descriptions to predict users’ preference over recipes.
- FM-VBPR adds visual features into FM framework above, which is projected into an embedding as the user ID, recipe ID, ingredients.

Note that the input for MF-BPR and LDA are users and recipes embeddings. Different from the above methods, FM adds ingredient embeddings as input features, VBPR adds visual features as input. Similar with our method HAFR, FM-VBPR considers all features as input, including users, recipes, ingredients and visual features. For hyperparameters tuning, We tune the learning rate and the coefficient for \( L_2 \) regularization for all models except LDA.

Parameter Settings. We implement our model and baseline methods in TensorFlow, which would be accessible via https://anonymous.com. For each method, we select the optimal hyper-parameters w.r.t. AUC on validation set. For a fair comparison, all models are set with an embedding size of 64 and optimized using the mini-batch Ada-grad (Duchi, Hazan, and Singer 2011) with a batch size of

![Figure 4: The distribution of user activity and recipe popularity over the Allrecipes dataset.](https://www.librec.net/)

- RQ3: How do the models perform with respect to the number of ratings per recipe, i.e., the popularity of recipe?
Table 1: Performance of compared methods. * denotes the statistical significance for \( p < 0.05 \).

| Methods    | Allrecipes AUC | NDCG@10 | Recall@10 |
|------------|----------------|---------|-----------|
| MF-BPR     | 0.5622         | 0.0376  | 0.0567    |
| LDA        | 0.5154         | 0.0376  | 0.0601    |
| FM         | 0.5710         | 0.0396  | 0.0607    |
| VBPR       | 0.5808         | 0.0296  | 0.0431    |
| FM-VBPR    | 0.5840         | 0.0372  | 0.0580    |
| HAFR-non-v | 0.6004         | 0.0332  | 0.0517    |
| HAFR-non-i | 0.6133         | 0.0418  | 0.0608    |
| HAFR       | 0.6435*        | 0.0455* | 0.0674*   |

Figure 5: Performance of Top-K recipe recommendation.

512. Besides, we search the learning rate within [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05] and \( \lambda \) for the regularization term among [0.0001, 0.001, 0.01, 0.1, 1]. Note that we use different \( \lambda \) for parameters of embeddings (\( P, Q, \) and \( X \)), mapping layer of image feature, and the MLP of output layer. Instead of considering both the two levels of attentions, basic HAFR uses a uniform weight to all entities. The proposed HAFR achieves good performance when the learning rate is 0.05 and regularization terms for embedding, image mapping layer, and MLP equal to [0.1, 0.01, 1], respectively. Without special mention, we report the performance of HAFR on this special setting.

4.2 Model Comparison (RQ1)

Table 1 displays the performance comparisons across different models. Note that HAFR-non-v and HAFR-non-i are two variants of HAFR without input of recipe image and ingredients, respectively. From Table 1, we have the following observations:

- The HAFR significantly outperforms other compared methods, e.g., better than FM-VBPR, the state-of-the-art model, by 12%. This justifies the utility of HAFR that leverages the strong representation capability of multi-layer neural network and the hierarchical attention to jointly model user-recipe interactions and recipe contents (i.e., image and ingredients).

- Compared to HAFR-non-v and HAFR-non-i, HAFR outperforms them by 24% and 8% on average, respectively. This illustrates that visual features and ingredient information are helpful for food recommendation because these side information can be informative and complement each other. This is also the reason why FM-VBPR, additionally incorporating visual features, performs better than FM.

- Among the baselines, FM-VBPR performs the best, which verifies the advantage of jointly modeling of historical interactions and recipe contents. VBPR and FM performs weaker since lacking ingredients and visual features, respectively. Besides, MF and LDA, modeling user-recipe interactions alone, performs further worse, which justifies the benefit of recipe context once again.

Note that NDCG and Recall metrics are not consistent with AUC metric. The reason may be that AUC is consistent with BPR loss, whereas NDCG and Recall focus more on recipe positions.

Figure 5 shows the performance of Top-K recipe recommendation where the ranking position \( K \) ranges from 1 to 20. As can be seen, HAFR consistently achieves the best performance on both NDCG and Recall metrics. This illustrates the robustness of HAFR and effectiveness of attention mechanism to explore information comprehensively. VBPR, as the only model considering both interactions and recipe image, performs unexpectedly bad, of which the reason is left for future exploration.

4.3 Effect of the Hierarchical Attention (RQ2)

To better understand the proposed HAFR model, we further evaluate its key component—the Hierarchical Attention, which consists of ingredient-level and component-level attention modules. Specifically, we compare two variants of HAFR by removing the ingredient-level attention (and the component-level one). It should be noted that the aggregation operation (e.g., aggregating ingredient embeddings into the representation of an ingredient list) will deteriorate to a common average pooling as removing the associated attention module. Table 2 shows the performance of HAFR and its variants, from which, we have the following key observations:

- When the component-level attention is applied, the performance is improved (by 4% on average) as compared with utilizing average pooling. The good performance signifies that users’ preference over recipes comes from different components (i.e., eating history, appearance, and ingredients), and justifies the effectiveness of dynamically aggregating the recipe information.

- By further applying the ingredient-level attention, relative performance improvement increases to 15%. The further performance improvement demonstrates the benefit of dynamically awarding ingredients, which provide clues to infer users’ preference over recipes. Besides, it indicates that an attention working at fine-grained (ingredient) level could benefit the attentive fusion at coarse-grained (component) level, which justifies the advantage of Hierarchical Attention.

4.4 Performance over Recipes of Different Sparsity Levels (RQ3)

Recall that the recipe popularity follows a long tail distribution. We then investigate the performance of models over
Table 2: Effect of attentions on ingredient (Ingre) and component (Comp) level. AVG and ATT represents the average pooling and attention, respectively. * denotes the statistical significance for $p < 0.05$.

| Model | Level | Allrecipes |
|-------|-------|------------|
| HAFR  | Ingre | 0.5927     |
|       | Comp  | 0.0373     |
|       | AVG   | 0.0576     |
|       | ATT   | 0.0582     |
|       | AVG   | 0.6215     |
|       | ATT   | 0.6435*    |
|       | A VG  | 0.0455*    |
|       | ATT   | 0.0674*    |

Figure 6: Performance over recipes of different sparsity.

recipes of different sparsity levels, i.e., we separately evaluate a trained model on recipes with different number of historical ratings. Specifically, we divide the test set into four groups with equal size by the number of ratings per recipe. Figure 6 illustrates the performance of the compared models, from which we observe:

- Our model HAFR with hierarchical attention consistently outperforms other baseline methods in all the groups. The improvement demonstrates the robustness and capacity of HAFR, which may be because HAFR learns better food representation with the hierarchical attention, which thoroughly explores the interaction and context information of the recipe.

- We can see a clear trend that the performance of the models increases with more ratings per recipe. The reason may be that for popular recipes, there are more user-recipe historical information. Therefore, the models could capture user preferences more easily.

5 Related Work

Food Recommendation. Food recommendation has received a lot of attentions in recent years. Technically speaking, existing work can be divided into two categories: collaborative filtering and content-based filtering approaches. Collaborative filtering approaches focus on mining users’ preference over recipes purely from user-recipe interactions with general CF algorithms, such as MF (Ge et al. 2015) and LDA (Trattner and Elsweiler 2017). However, CF-based methods ignore the rich content of recipes, which is of vital importance for food recommendation.

Distinct from collaborative filtering approaches, content-based filtering methods only explore food contents for making recommendation. Regarding the types of considered contents, researches in this line can be further subdivided into ingredient-based, image-based, and profile-based ones.

1) Ingredient-based methods make recommendations by mining user preference from recipe ingredients (Freine and Berkovsky 2010a, Freine and Berkovsky 2010b, Teng, Lin, and Adamic 2012). For instance, Teng et al. (2012) take ingredients as recipe features, which are fed into a tree-based classifier to predict rating on a recipe from the target user.

2) Image-based methods show that algorithms designed to extrapolate important visual aspects of recipe images would enhance food recommendation (Yang et al. 2015, Yang et al. 2017). Besides recipe contents, some studies show that side information of user, including gender (Rokicki et al. 2016), cultural background (Kim and Chung 2016) and other demographic factors (Weber and Acharanuparp 2016, Rokicki, Herder, and Trattner 2017), also reflects users’ characteristics over food choices. However, none of the existing methods jointly model the user-recipe interactions and key contents of recipe.

Visually-aware Recommendation. With the development of image processing techniques, great efforts from both the industry and academia have been paid on visually-aware recommendation in two main directions: recommending visual contents and enhancing recommender system with visual contents. Researches in the former direction largely view image features as item contents and mainly focus on improving user representation with side information (Geng et al. 2015, Lei et al. 2016, Chen et al. 2017, Gelli et al. 2017). Our work fall into the later direction, where visual contents are viewed as side information of the item to be recommended, such as, point-of-interests (POIs) (Wang et al. 2017) and fashion products (e.g., clothes and accessory) (Yu et al. 2018). Different from these items, food also has key content information of its ingredients, which is closely related to the appearance of food. As such, we devise a hierarchical attention, which carefully aggregates these contents to improve recommendation service.

6 Conclusion

In this work, we proposed Hierarchical Attention based Food Recommendation (HAFR) to infer users’ preference over recipes for food recommendation. The HAFR aims to learn more comprehensive recipe representation via jointly leveraging user-recipe interaction history, food image, and food ingredients with a hierarchical attention. We collected a large-scale dataset for food recommendation and conducted extensive experiments, demonstrating that HAFR consistently outperforms the state-of-the-art models. Besides, the ablation experiments demonstrate the usefulness of aggregating recipe information in a hierarchical fashion.

In future, we are interested in exploring the following directions: 1) we plan to incorporate healthy and nutrition factors into the food recommendation framework, so that the recommender would guide users to a healthier eating style. 2) We will explore the relations (e.g., replaceable and complementary relations) between ingredients to further enhance the recipe representation. And 3) We will incorporate side information of users into the food recommendation framework to further improve its performance.
[Yang et al. 2016] Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In NAACL, 1480–1489.

[Yang et al. 2017] Yang, L.; Hsieh, C.-K.; Yang, H.; Pollak, J. P.; Dell, N.; Belongie, S.; Cole, C.; and Estrin, D. 2017. Yum-me: a personalized nutrient-based meal recommender system. ACM Transactions on Information Systems (TOIS) 36(1):7.

[Yu et al. 2018] Yu, W.; Zhang, H.; He, X.; Chen, X.; Xiong, L.; and Qin, Z. 2018. Aesthetic-based clothing recommendation. In WWW, 649–658.