Recipe2Vec: Multi-modal Recipe Representation Learning with Graph Neural Networks

Yijun Tian[1], Chuxu Zhang[2*], Zhichun Guo[1], Yihong Ma[1], Ronald Metoyer[1], Nitesh V. Chawla[1*]

[1]Department of Computer Science, University of Notre Dame, USA
[2]Department of Computer Science, Brandeis University, USA

[1]{yijun.tian, zguo5, yma5, rmetoyer, nchawla}@nd.edu, [2]chuxuzhang@brandeis.edu

Abstract

Learning effective recipe representations is essential in food studies. Unlike what has been developed for image-based recipe retrieval or learning structural text embeddings, the combined effect of multi-modal information (i.e., recipe images, text, and relation data) receives less attention. In this paper, we formalize the problem of *multi-modal recipe representation learning* to integrate the visual, textual, and relational information into recipe embeddings. In particular, we first present *Large-RG*, a new recipe graph data with over half a million nodes, making it the largest recipe graph to date. We then propose *Recipe2Vec*, a novel graph neural network based recipe embedding model to capture multi-modal information. Additionally, we introduce an adversarial attack strategy to ensure stable learning and improve performance. Finally, we design a joint objective function of node classification and adversarial learning to optimize the model. Extensive experiments demonstrate that *Recipe2Vec* outperforms state-of-the-art baselines on two classic food study tasks, i.e., cuisine category classification and region prediction. Dataset and codes are available at https://github.com/meettyj/Recipe2Vec.

1 Introduction

Large-scale food data provides a wealth of information about food and can assist in resolving many critical societal issues [Min *et al.*, 2019]. Recipe representation learning, in particular, embeds recipes in a latent space, allowing for the extraction of hidden information from massive food datasets and facilitating various application tasks that rely on a robust food space representation, such as culinary habits analysis [Min *et al.*, 2018], recipe healthiness estimation [Rokicki *et al.*, 2018], recipe recommendation [Tian *et al.*, 2022b], and recipe classification [Li and Zaki, 2020].

Existing approaches to recipe representation learning rely entirely on textual content [Li and Zaki, 2020]. They
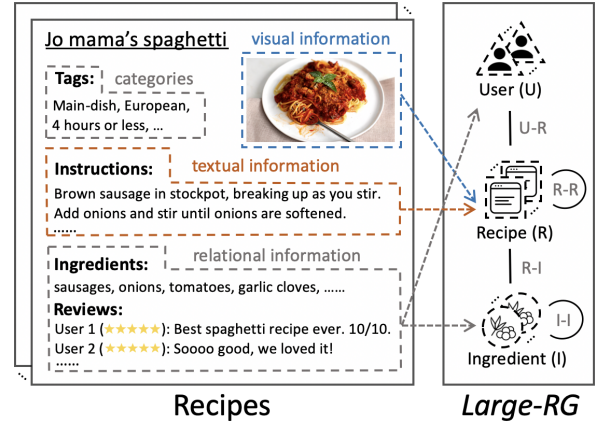
---
*Corresponding authors

Figure 1: The illustration of *Large-RG*. We associate visual and textual information with recipe nodes, and extract various types of relations among user, recipe, and ingredient nodes.

take advantage of the recipe ingredients and instructions but overlook the relationship between recipes and food items. Besides leveraging the textual information, rn2vec [Tian *et al.*, 2021] proposes a recipe network embedding model to capture the relational information. However, rn2vec ignores the large number of images associated with each recipe and thus results in suboptimal performance. While some studies consider recipe images, they typically focus on image-based applications such as recognizing and retrieving recipes from images [Pandey *et al.*, 2017; Chen *et al.*, 2018]. These works attempt to align the images and the text in a shared embedding space for specific downstream tasks, but ignore the unique information contained in each modality.

In this paper, we propose the problem of *multi-modal recipe representation learning*, which leverages different modalities such as images, text, and relations to learn recipe embeddings. To solve this problem, we first create and release a new recipe graph, *Large-RG*, which contains over half a million nodes and is the largest recipe graph to date. As shown in Fig. 1, we extract visual, textual, and relational information of each recipe to build *Large-RG*, which consists of three types of nodes and four types of edges. We then propose *Recipe2Vec*, a novel graph neural network (GNN) based model for multi-modal recipe representation

learning. Specifically, we first encode the node attributes using various pre-trained neural networks, including a two-stage LSTM [Marin *et al.*, 2019] for instructions and a ResNet [He *et al.*, 2016] for images. Next, we introduce a multi-view neighbor sampler to capture both local and high-order information from graph schema-based and meta-path-based neighbors, respectively. We also design several neural network based modules to fuse the information from different nodes, modalities, and relations. In addition, we introduce a feature-based adversarial attack strategy to ensure stable learning and improve model performance. Finally, we design a novel combined objective function of node classification and adversarial learning to optimize the model. To summarize, our major contributions are as follows:

- As the first attempt to study the problem of *multi-modal recipe representation learning*, we create and release *Large-RG*, a new recipe graph data with over half a million nodes, which is the largest recipe graph to date.

- We propose a novel GNN-based recipe embedding model *Recipe2Vec* to solve the problem. *Recipe2Vec* is able to capture visual, textual, and relational information and learn effective representations through several neural network modules. We further introduce a novel objective function to optimize the model.

- We conduct extensive experiments to evaluate the performance of our model. The results show the superiority of *Recipe2Vec* by comparing it with state-of-the-art baselines on two classic food study tasks: cuisine category classification and region prediction.

2 Related Work

This work is closely related to recipe datasets, recipe representation learning, and graph neural networks.

Recipe Datasets. Existing datasets focus exclusively on either recipe images, text, or the simple relationship between ingredients [Marin *et al.*, 2019; Haussmann *et al.*, 2019; Tian *et al.*, 2021; Tian *et al.*, 2022a], but fail to contain all of these information and further neglect the complex structure or user information, resulting in poor generality and integrity. Different from existing works, we create and release a recipe graph data which contains extensive relational information and multi-modal information including images and text.

Recipe Representation Learning. Existing works focus on using textual content to learn recipe representations [Li and Zaki, 2020; Tian *et al.*, 2021]. They take advantage of the instructions and ingredients associated with recipes but ignore the large number of recipe images. Several other works consider the images, [Carvalho *et al.*, 2018; Marin *et al.*, 2019; Wang *et al.*, 2019a; Salvador *et al.*, 2021] but they typically concentrate on the recipe-image retrieval task and attempt to align images and text together in a shared space, resulting in information loss for both modalities. Instead, we focus on learning recipe representations using multi-modal information extracted from images, text, and relations.

Graph Neural Networks. Many GNNs [Veličković *et al.*, 2018; Schlichtkrull *et al.*, 2018; Wang *et al.*, 2019b; Zhang *et al.*, 2019a; Zhang *et al.*, 2019b] were proposed to

Component	Name	Number
Nodes	user	38,624
	recipe	472,515
	ingredient	22,186
Edges	user-recipe	1,193,179
	recipe-recipe	644,256
	recipe-ingredient	4,440,820
	ingredient-ingredient	170,642

Table 1: The statistics of *Large-RG*.

learn vectorized node embeddings. They take advantage of both node attributes and relational information to learn node embeddings for different graph mining tasks. For example, GAT [Veličković *et al.*, 2018] employs self-attention mechanisms to measure the impacts of different neighbors and combine their effects to obtain node embeddings. HAN [Wang *et al.*, 2019b] uses hierarchical attentions to learn embeddings by aggregating information from both node-level and semantic-level structures. Inspired by these studies, we build a GNN-based model to learn recipe embeddings. In addition, adversarial learning has shown great performance in training neural networks [Jiang *et al.*, 2020; Xie *et al.*, 2020]. We thus introduce a feature-based adversarial attack strategy to ensure stable learning and further improve performance.

3 Preliminary

In this section, we describe the concept of heterogeneous recipe graph and formally define the problem of *multi-modal recipe representation learning*. We further introduce the *Large-RG* data that we create in this work.

Definition 3.1. Heterogeneous Recipe Graph. *A heterogeneous recipe graph is defined as a graph $G = (V, E, X)$ with multiple types of nodes V (i.e., User, Recipe, Ingredient) and edges E (e.g., U-R, R-R, R-I, I-I in Fig. 1). In addition, nodes are associated with attributes X, e.g., images, instructions, and nutrients.*

Problem 1. Multi-modal Recipe Representation Learning. *Given a recipe graph $G = (V, E, X)$, the task is to design a learning model $\mathcal{F}_\Theta$ with parameters Θ to learn d-dimensional recipe embeddings $\mathcal{E} \in \mathbb{R}^d$, while encoding the multi-modal information (i.e., visual, textual, and relational information). The learned recipe representations can be utilized in various downstream tasks such as cuisine category classification and region prediction.*

Large-RG Data. To solve the problem, we first create *Large-RG* data. The statistics of *Large-RG* are reported in Tab. 1. Specifically, we first collect recipes from Recipe1M [Marin *et al.*, 2019] and crawl the user ratings for each recipe from *food.com*. We then match each ingredient to the USDA nutritional database [USDA, 2019] to determine its nutritional value. After that, we build *Large-RG* by representing users, recipes, and ingredients as nodes and connecting them with four types of relations. In particular, we build the recipe-ingredient relation by connecting each recipe and its ingredients. We construct the user-recipe relation based on the user ratings. We also connect recipe nodes by the similarity determined by FoodKG [Haussmann
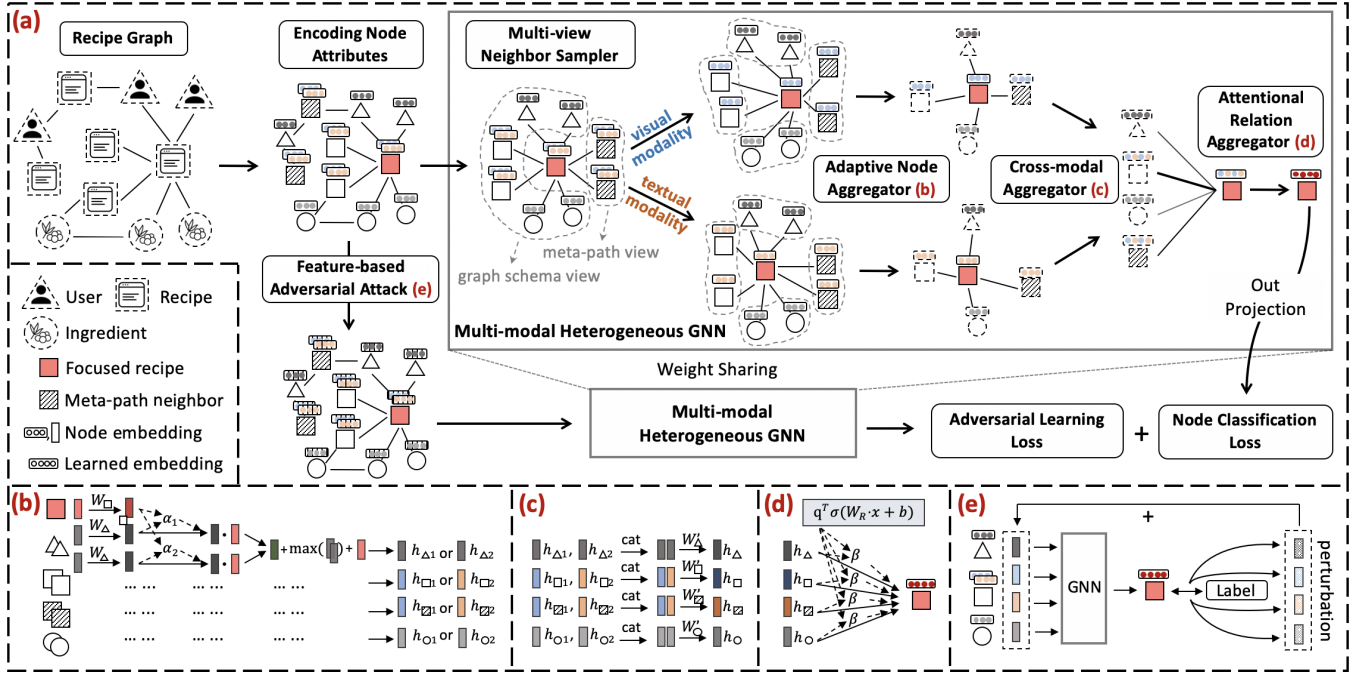
Figure 2: (a) The overall framework of *Recipe2Vec*: we first encode the node attributes and propose a multi-modal heterogeneous GNN with various modules to encode visual, textual, and relational information for learning recipe embeddings. We then introduce a feature-based adversarial attack strategy to ensure stable learning and design a joint loss of node classification and adversarial learning to optimize the model; (b) Adaptive node aggregator: encoding information from nodes under a specific relation; (c) Cross-modal aggregator: fusing information from different modalities; (d) Attentional relation aggregator: using attention mechanism to fuse embeddings over different relations; (e) Feature-based adversarial attack: learning adversarial features by generating adversarial perturbations for input features.

et al., 2019] and connect ingredient nodes based on the co-occurring probabilities. Additionally, we crawl recipe tags from food.com to create two categories and assign each recipe a cuisine class and a region class [Tian *et al.*, 2021].

4 Model

In this section, we present the details of *Recipe2Vec*. As illustrated in Fig. 2, we first encode the node attributes and develop a multi-modal heterogeneous GNN to learn recipe embeddings. The proposed GNN contains several modules including a multi-view neighbor sampler, an adaptive node aggregator (Fig. 2(b)), a cross-modal aggregator (Fig. 2(c)), and an attentional relation aggregator (Fig. 2(d)). We also introduce a feature-based adversarial attack strategy (Fig. 2(e)) to ensure stable learning. Finally, we design a novel objective function to optimize the model.

4.1 Encoding Nodes Attributes

We pretrain a ResNet [He *et al.*, 2016] to encode the recipe images and a two-stage LSTM [Marin *et al.*, 2019] to encode the instructions. Next, we associate the pre-trained image embeddings (x_{img}) and instruction embeddings (x_{ins}) with recipe nodes as the node attributes. Similarly, we consider nutrients as the attributes of ingredient nodes (x_{ing}) and random vectors as the attributes of user nodes (x_{user}). Due to the various modality of attributes, we introduce a modality-specific input projection to project the node attributes into a shared embedding space:

$$h_{i,m} = W_m \cdot x_{i,m}, \quad (1)$$

where $W_m \in \mathbb{R}^{d_m \times d}$ is the projection for modality m, $x_{i,m}$ is the associated attribute of node v_i with dimension d_m, and $h_{i,m}$ is the projected feature of node v_i with dimension d.

4.2 Multi-modal Heterogeneous GNN

For each recipe node v_i with input feature $h_{i,m}$, we first introduce a multi-view neighbor sampler to sample nodes from graph schema view and meta-path view. We then use an adaptive node aggregator to encode the information from nodes under a specific relation. Next, we propose a cross-modal aggregator to fuse the information from different modalities. After that, we design an attentional relation aggregator to combine the information from different relations and obtain the final recipe embedding.

Multi-view Neighbor Sampler. We introduce the multi-view neighbor sampler to simultaneously capture local information through graph schema view and high-order information through meta-path view [Dong *et al.*, 2017]. In the graph schema view, the node aggregates the information from its direct neighbors, which we denote as the graph schema-based neighbors N_i^S. In the meta-path view, the neighbors are determined by meta-path walks for the high-order information. In particular, given a meta-path $\mathcal{P}$ that starts from v_i, instead of taking all the nodes that appear in the meta-path as neighbors, we define an importance

pooling to select the most important p nodes. We start by simulating multiple meta-path walks from v_i and compute the L_1 normalized visit count for each node. We then determine the important neighbors by choosing the top p nodes with the highest normalized visit counts. We denoted these selected important neighbors as the meta-path-based neighbors N_i^P. Finally, the neighbor nodes of v_i is the combined set of N_i^S and N_i^P, denoted as N_i. In addition, since the recipe nodes are associated with different modalities (e.g., visual and textual), we split the graph into modality-specific graphs to account for each modality individually.

Adaptive Node Aggregator. Given a node v_i and its neighbor nodes N_i, we first sample the relation-specific neighbors $N_{i,r} \subset N_i$ by choosing the nodes that connect to v_i through relation r. Then, we calculate the unnormalized attention score $e_{ij,m}$ between v_i and $v_j \in N_{i,r}$ given the modality m, and normalize the $e_{ij,m}$ using $softmax$ function to make the coefficients easily comparable:

$$e_{ij,m} = \text{LeakyReLU}\left[W_{ij} \cdot (h_{i,m} \| h_{j,m})\right],$$
$$\alpha_{ij,m} = \frac{\exp(e_{ij,m})}{\sum_{k \in N_{i,r}} \exp(e_{ik,m})}, \qquad (2)$$

where $\|$ indicates concatenation, $W_{ij} \in \mathbb{R}^{2d \times d}$ is a shared weight vector, and $\alpha_{ij,m}$ is the attention weight between v_i and v_j for modality m. We then use the $\alpha_{ij,m}$ as coefficients to linearly combine the interaction and yield the united interaction-dependent affinity A_m for each modality m:

$$A_m = \sum_{j \in N_{i,r}} \alpha_{ij,m} \cdot W_a \cdot (h_{i,m} \odot h_{j,m}), \qquad (3)$$

where $W_a \in \mathbb{R}^{d \times d}$ is a shared trainable weight and $\odot$ denotes the element-wise product. We then aggregate the neighbor node features following GIN [Xu *et al.*, 2019] while including two shared transformation matrix $W_i \in \mathbb{R}^{d \times d}$ and $W_j \in \mathbb{R}^{d \times d}$ to enhance learning capacity. We further make it adaptive to specific downstream tasks by incorporating the interaction-dependent affinity A_m, which makes the message dependent on the affinity between v_i and v_j:

$$h_{i,r,m} = W_o \cdot [W_i \cdot h_i + \max(\sum_{j \in N_{i,r}} W_j \cdot h_j) + A_m], \quad (4)$$

where $h_{i,r,m}$ is the encoded embedding of v_i through relation r for modality m, and $W_o \in \mathbb{R}^{d \times d}$ is a shared weight vector.

Cross-modal Aggregator. To better exploit the information in different modalities and learn a joint embedding, we design the cross-modal aggregator to fuse embeddings from visual and textual modalities. Specifically, we first concatenate the $h_{i,r,image}$ and $h_{i,r,text}$, and then introduce a node type-specific projection $W_{\phi_i} \in \mathbb{R}^{2d \times d}$ to transform the concatenated embedding, with ϕ_i are the node type of v_i:

$$h_{i,r} = W_{\phi_i} \cdot (h_{i,r,image} \| h_{i,r,text}), \qquad (5)$$

where $h_{i,r}$ is the learned cross-modal embedding of node v_i through relation r.

Attentional Relation Aggregator. We further introduce the attentional relation aggregator to learn the importance of each relation and fuse them accordingly. Specifically, we first apply a weight matrix $W_R \in \mathbb{R}^{d \times d}$ to transform the $h_{i,r}$ and use a relation-level vector $q \in \mathbb{R}^d$ to calculate the similarity. We then average the similarity scores of all nodes that connected to relation r to obtain the importance score $w_{i,r}$ for node v_i. The process is formulated as follows:

$$w_{i,r} = \frac{1}{|V_r|} \sum_{i \in V_r} q^{\text{T}} \cdot \tanh(W_R \cdot h_{i,r} + b), \qquad (6)$$

where V_r denotes the set of nodes that are connected to relation r, and $b \in \mathbb{R}^d$ is the bias vector. Next, we normalize $w_{i,r}$ to get the final relation-level attention weight $\beta_{i,r}$:

$$\beta_{i,r} = \frac{\exp(w_{i,r})}{\sum_{r \in R_i} \exp(w_{i,r})}, \qquad (7)$$

where R_i indicates the associated relations of v_i. After that, we fuse the cross-modal embedding $h_{i,r}$ with $\beta_{i,r}$ to obtain the final recipe embedding h_i:

$$h_i = \sum_{r=1}^{R_i} \beta_{i,r} \cdot h_{i,r}. \qquad (8)$$

4.3 Objective Function

The learned recipe embedding h_i can be used for various downstream tasks such as cuisine category classification, region prediction, or recipe recommendation. Specifically, in this work, we can introduce a supervised node classification loss (e.g., cross-entropy) to classify the cuisine categories (or predict the recipe regions):

$$L_{sup} = - \sum_{i \in \mathcal{Y}_T} Y_i \log(FC(h_i)), \qquad (9)$$

where $\mathcal{Y}_T$ is the set of training data, FC denotes the fully connected layer, and Y_i is the one-hot label of v_i.

4.4 Feature-based Adversarial Learning

Since the above objective function only considers the explicit feature information associated with each node, we further introduce an adversarial attack strategy to leverage the regularization power of adversarial features. Specifically, we choose Projected Gradient Descent [Madry *et al.*, 2018] as the default attacker to generate adversarial features on the fly. Compared to the vanilla training that we send original input features $h_{i,m}$ (Eq. 1) into GNN to obtain the learned recipe embedding h_i (Eq. 8), adversarial training takes the maliciously perturbed features $h'_{i,m}$ as input and obtains the perturbed recipe embedding h'_i, which is further used to calculate the adversarial learning loss L_{adv}:

$$h'_{i,m} = h_{i,m} + \epsilon_m; h'_i = GNN(h'_{i,m}),$$
$$L_{adv} = \max_{\epsilon_m \in \mathbb{S}} [- \sum_{i \in \mathcal{Y}_T} Y_i \log(FC(h'_i))], \qquad (10)$$

where ϵ_m is the adversarial perturbation for modality m and $\mathbb{S}$ is the allowed perturbation range. The final objective function L is defined as the weighted combination of L_{sup} and L_{adv}:

$$L = L_{sup} + \lambda L_{adv}, \qquad (11)$$

where λ is a trade-off weight for balancing two losses.

Metric	Method	Appetizer	Beverage	Bread	Soups	Salads	Desserts	Vegetables	Main-dish	Others	Total
F1	TextCNN	50.4	85.9	73.8	62.3	68.9	86.9	49.1	75.9	42.4	73.5
	ResNet	41.9	88.5	74.3	75.9	76.8	88.7	64.3	83.4	55.6	78.7
	mp2vec	50.0	83.2	67.3	57.7	63.2	85.9	54.9	75.7	43.0	72.4
	RGCN	52.6	81.8	65.3	52.5	60.1	84.0	49.4	73.7	43.5	70.7
	GAT	53.2	84.7	60.7	51.2	64.9	84.4	38.5	73.5	41.8	71.0
	GIN	68.4	<u>91.9</u>	82.4	81.2	81.4	91.4	70.9	86.1	70.3	83.9
	HAN	67.0	90.4	78.3	79.1	78.9	89.4	66.1	84.6	67.3	81.8
	HGT	49.1	89.3	75.1	79.1	78.0	88.4	63.5	82.8	55.3	79.2
	Reciptor	68.2	89.1	79.3	77.4	71.4	90.2	59.9	84.1	57.4	80.9
	rn2vec	<u>76.8</u>	91.4	<u>84.4</u>	<u>83.7</u>	<u>83.9</u>	<u>91.5</u>	<u>76.8</u>	<u>86.9</u>	<u>73.5</u>	<u>85.9</u>
	Recipe2Vec	**82.3**	**95.5**	**89.0**	**87.9**	**87.2**	**94.5**	**82.3**	**89.4**	**78.2**	**89.4**
Acc	TextCNN	45.5	86.3	74.2	61.0	70.9	90.8	41.8	85.8	35.2	69.3
	ResNet	32.0	89.8	72.9	67.9	72.9	92.1	59.9	90.3	56.6	74.9
	mp2vec	40.7	87.8	61.3	68.4	61.8	90.1	50.4	86.4	34.7	68.6
	RGCN	47.3	78.7	57.8	43.8	58.0	92.1	44.6	83.7	37.1	66.9
	GAT	44.4	81.4	47.6	39.6	70.3	94.6	26.0	84.1	40.0	66.4
	GIN	66.0	<u>93.8</u>	79.7	78.3	81.4	92.0	65.6	89.7	69.5	81.7
	HAN	62.0	91.7	75.2	75.1	80.4	92.2	61.6	89.1	64.3	79.5
	HGT	40.7	89.4	70.4	75.7	75.9	93.4	54.6	90.8	53.2	75.5
	Reciptor	62.7	86.2	76.2	75.4	71.3	93.6	55.0	91.2	52.7	77.4
	rn2vec	<u>72.5</u>	90.4	<u>81.3</u>	<u>80.3</u>	<u>82.7</u>	<u>94.3</u>	<u>72.9</u>	<u>91.4</u>	<u>70.5</u>	<u>84.0</u>
	Recipe2Vec	**80.1**	**95.8**	**87.5**	**87.5**	**88.8**	**95.5**	**80.1**	**91.6**	**76.0**	**87.6**

Table 2: Cuisine category classification results.

Metric	Method	American	European	Asian	Mexican	Total
F1	TextCNN	67.8	56.7	64.3	37.8	62.2
	ResNet	69.7	58.1	66.7	40.1	64.1
	mp2vec	70.2	64.0	<u>81.4</u>	20.1	67.7
	RGCN	70.3	64.1	70.7	30.3	66.4
	GAT	69.4	63.6	80.3	24.3	67.0
	GIN	73.7	65.5	75.7	55.4	70.2
	HAN	71.8	66.8	78.7	58.3	70.2
	HGT	71.8	65.5	78.1	50.5	69.5
	Reciptor	73.2	66.8	76.2	57.2	70.6
	rn2vec	<u>75.6</u>	<u>68.5</u>	79.6	<u>59.8</u>	<u>73.0</u>
	Recipe2Vec	**78.7**	**73.8**	**85.3**	**68.9**	**77.5**
Acc	TextCNN	72.3	57.0	60.0	28.0	62.2
	ResNet	75.8	55.8	63.2	30.1	64.1
	mp2vec	75.4	65.9	77.5	11.5	67.7
	RGCN	76.7	68.6	57.1	18.8	66.4
	GAT	73.5	68.2	73.0	14.3	67.0
	GIN	77.1	66.2	69.3	48.2	70.2
	HAN	71.7	73.0	70.7	50.7	70.2
	HGT	75.2	68.5	70.5	38.9	69.5
	Reciptor	72.6	<u>69.4</u>	<u>77.8</u>	49.7	70.6
	rn2vec	<u>78.1</u>	67.4	<u>77.8</u>	55.0	<u>73.0</u>
	Recipe2Vec	**78.6**	**74.0**	**83.8**	**71.0**	**77.5**

Table 3: Region prediction results.

5 Experiments

In this section, we conduct extensive experiments to evaluate the performance of *Recipe2Vec* and show related analyses.

5.1 Baseline Methods

We compare with 10 baselines including the classic classifiers **TextCNN** [Kim, 2014], **ResNet** [He *et al.*, 2016], homogeneous graph embedding models **GAT** [Veličković *et al.*, 2018], **GIN** [Xu *et al.*, 2019], heterogeneous graph embedding models **mp2vec** [Dong *et al.*, 2017], **RGCN** [Schlichtkrull *et al.*, 2018], **HAN** [Wang *et al.*, 2019b], **HGT** [Hu *et al.*, 2020], and recipe representation learning models **Reciptor** [Li and Zaki, 2020] and **rn2vec** [Tian *et al.*, 2021].

5.2 Implementation Details

We split the data into train/validation/test set by 70/15/15. For the proposed *Recipe2Vec*, we set the learning rate to 0.005, the hidden size to 128, the input dimension of instruction and image embeddings to 512, the input dimension of ingredient embeddings to 46, batch size to 4096, meta-path $\mathcal{P}$ to recipe-user-recipe, the number of meta-path neighbors p to 10, training epochs to 100, the trade-off factor λ to 0.1, the perturbation range $\mathbb{S}$ to 0.02, number of iterations for attack to 5, and the attack step size to 0.005.

5.3 Performance Comparison

We use Micro-F1 and Accuracy (Acc) as the evaluation metrics and report the performances on cuisine category classification and region prediction tasks in Tab. 2 and Tab. 3, respectively. The best and second-best values are highlighted by bold and underline. According to these tables, we can find that *Recipe2Vec* outperforms all the baselines for both tasks in all cases. Specifically, classic classifiers (i.e., TextCNN and ResNet) perform poorly because of the neglect of both relational information and multi-modal information. Graph embedding methods (e.g., GIN and HAN) obtain decent performance after incorporating the complex relational information. Similarly, recipe representation learning models (e.g., rn2vec) achieve satisfactory results, but they fail to encode the influence of different modalities. Finally, *Recipe2Vec* achieves the best performance compared to all the baselines, by improving **+3.5%** (F1) and **+3.6%** (Acc) in cuisine category classification, and **+4.5%** (F1 and Acc) in region prediction. This demonstrates that *Recipe2Vec* can obtain better recipe embeddings compared to other models.

5.4 Ablation Study

Since *Recipe2Vec* contains various essential components (i.e., multi-view neighbor sampler (NS), adaptive node aggregator (NA), cross-modal aggregator (CA), attentional

Recipe Title	Ingredients	Ground Truth	TextCNN	HAN	rn2vec	*Recipe2Vec*
Corn and Coriander Soup	corn, capsicum, onion, coriander, ...	Soups	Main-dish	Soups	Soups	Soups
Herbed Tomatoes	tomatoes, herb, onion, rosemary, ...	Vegetables	Main-dish	Main-dish	Vegetables	Vegetables
Zucchini Spoon Bread	zucchini, eggs, milk, almonds, ...	Bread	Vegetables	Main-dish	Vegetables	Bread
Spaghetti With Pea Sauce	spaghetti, butter, shallots, peas, ...	Main-dish	Appetizer	Salads	Salads	Main-dish

Table 5: Error cases for cuisine category classification.

Task	Category	– NS	– NA	– CA	– RA	– AL	*Recipe2Vec*
	Appetizer	81.9	79.0	78.5	81.5	81.2	**82.3**
	Beverage	95.4	94.4	94.0	95.0	95.1	**95.5**
	Bread	88.6	87.4	86.9	88.6	88.3	**89.0**
Cuisine	Soups	87.5	86.1	86.0	87.8	87.7	**87.9**
category	Salads	86.5	85.6	85.7	86.5	86.5	**87.2**
classification	Desserts	94.1	93.2	92.9	94.2	94.1	**94.5**
	Vegetables	81.3	78.2	78.6	81.9	81.1	**82.3**
	Main-dish	89.1	87.6	87.7	89.3	89.0	**89.4**
	Others	77.3	75.2	75.9	78.1	78.0	**78.2**
	Total	88.9	87.4	87.3	89.1	88.8	**89.4**
	American	78.5	77.3	78.2	78.4	78.5	**78.7**
Region	European	73.0	72.0	66.7	73.7	73.3	**73.8**
prediction	Asian	84.3	83.1	82.9	84.4	83.7	**85.3**
	Mexican	65.6	64.4	64.5	67.2	67.1	**68.9**
	Total	76.8	75.7	75.1	77.1	76.9	**77.5**

Table 4: F1 scores of different model variants.

relation aggregator (RA), and feature-based adversarial learning (AL)), we conduct ablation studies to analyze the contributions of different components by removing each of them independently (see Tab. 4). Specifically, removing NA and CA significantly affects the performance, showing that both NA and CA have large contributions to *Recipe2Vec*. In addition, we remove NS, RA, and AL from our model, respectively. The decreasing performance of these model variants demonstrates the effectiveness of NS, RA, and AL in enhancing the model. Finally, *Recipe2Vec* achieves the best results in all cases, indicating the strong capability of different components in our model.

5.5 Case Study

To show the performance of different models with concrete examples, we analyze the misclassified cases in the cuisine category classification task, as shown in Tab. 5. Specifically, TextCNN misclassifies all of these four recipes, indicating that the textual features cannot fully represent the recipes and that ignoring relational information may lead to suboptimal performance. HAN successfully classifies the recipe *Corn and Coriander Soup*, but fails to classify the others. One potential reason is that relying solely on meta-path-based neighbors may result in information loss. rn2vec successfully classifies the recipes *Corn and Coriander Soup* and *Herbed Tomatoes*, but fails to classify the other two. This may be because the model cannot fully capture the multi-modal information by only using a simple GNN-based structure. However, our model *Recipe2Vec* takes into account the visual, textual, and relational information through several neural network modules, which clearly distinguishes the difference among categories and correctly classifies these recipes.
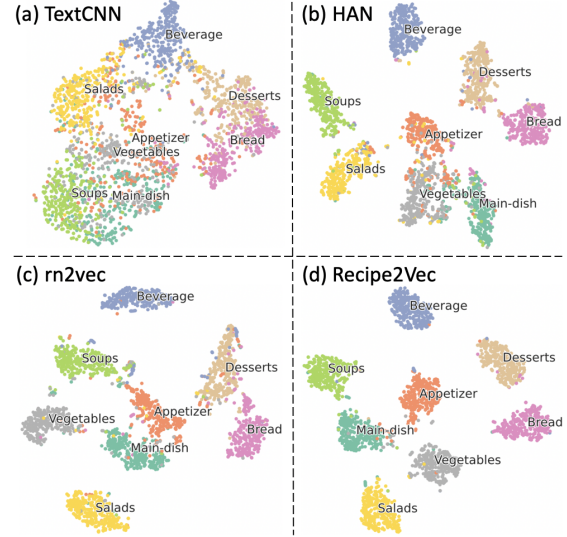

Figure 3: Embedding visualization of recipes.

5.6 Embedding Visualization

For a more intuitive understanding and comparison, we visualize embeddings of different models using t-SNE. As shown in Fig. 3, TextCNN does not perform well. Only Beverage, Salads, Desserts, and Bread are separated apart while other categories are mixed. HAN can separate most categories but fails to distinguish the Appetizer, Vegetables, and Main-dish. While rn2vec can successfully separate all categories, their distinctions are obscure, i.e., points from different categories are close to each other. However, our model *Recipe2Vec* can clearly identify each category and distinctly separate all of them. This again demonstrates that *Recipe2Vec* can learn discriminative recipe embeddings.

6 Conclusion

In this paper, we propose and formalize the problem of *multi-modal recipe representation learning*. To solve this problem, we create and release *Large-RG*, a new and large-scale recipe graph data to facilitate graph-based food studies. Furthermore, we develop *Recipe2Vec*, a novel GNN-based recipe embedding model. *Recipe2Vec* is able to capture visual, textual, and relational information through various carefully designed neural network modules. We also design a joint objective function of node classification and adversarial learning to optimize the model. Extensive experiments show that *Recipe2Vec* outperforms state-of-the-art baselines on two classic food study tasks.

Acknowledgements

References

[Carvalho et al., 2018] Micael Carvalho, Rémi Cadène, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings. In *SIGIR*, 2018.

[Chen et al., 2018] Jing-Jing Chen, Chong-Wah Ngo, Fu-Li Feng, and Tat-Seng Chua. Deep understanding of cooking procedure for cross-modal recipe retrieval. In *MM*, 2018.

[Dong et al., 2017] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, 2017.

[Haussmann et al., 2019] Steven Haussmann, Oshani Seneviratne, Yu Chen, Yarden Ne'eman, James Codella, Ching-Hua Chen, Deborah L. McGuinness, and Mohammed J. Zaki. Foodkg: A semantics-driven knowledge graph for food recommendation. In *ISWC*, 2019.

[He et al., 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Hu et al., 2020] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *WWW*, 2020.

[Jiang et al., 2020] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. In *NeurIPS*, 2020.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[Li and Zaki, 2020] Diya Li and Mohammed J. Zaki. Reciptor: An effective pretrained model for recipe representation learning. In *KDD*, 2020.

[Madry et al., 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[Marin et al., 2019] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *TPAMI*, 2019.

[Min et al., 2018] Weiqing Min, Bing-Kun Bao, Shuhuan Mei, Yaohui Zhu, Yong Rui, and Shuqiang Jiang. You are what you eat: Exploring rich recipe information for cross-region food analysis. *IEEE Trans. Multimed.*, 2018.

[Min et al., 2019] Weiqing Min, Shuqiang Jiang, Linhu Liu, Yong Rui, and Ramesh Jain. A survey on food computing. *arXiv preprint arXiv:1808.07202*, 2019.

[Pandey et al., 2017] Paritosh Pandey, Akella Deepthi, Bappaditya Mandal, and Niladri B. Puhan. Foodnet: Recognizing foods using ensemble of deep networks. *IEEE Signal Processing Letters*, 2017.

[Rokicki et al., 2018] M. Rokicki, C. Trattner, and E. Herder. The impact of recipe features, social cues and demographics on estimating the healthiness of online recipes. In *ICWSM*, 2018.

[Salvador et al., 2021] Amaia Salvador, Erhan Gundogdu, Loris Bazzani, and Michael Donoser. Revamping cross-modal recipe retrieval with hierarchical transformers and self-supervised learning. In *CVPR*, 2021.

[Schlichtkrull et al., 2018] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *ESWC*, 2018.

[Tian et al., 2021] Yijun Tian, Chuxu Zhang, Ronald Metoyer, and Nitesh V. Chawla. Recipe representation learning with networks. In *CIKM*, 2021.

[Tian et al., 2022a] Yijun Tian, Chuxu Zhang, Zhichun Guo, Chao Huang, Ronald Metoyer, and Nitesh V. Chawla. Reciperec: A heterogeneous graph learning model for recipe recommendation. In *IJCAI*, 2022.

[Tian et al., 2022b] Yijun Tian, Chuxu Zhang, Ronald Metoyer, and Nitesh V. Chawla. Recipe recommendation with hierarchical graph attention network. *Frontiers in Big Data*, 2022.

[USDA, 2019] USDA. National nutrient database for standard reference, release 27, 2019.

[Veličković et al., 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Wang et al., 2019a] Hao Wang, Doyen Sahoo, Chenghao Liu, Ee-Peng Lim, and Steven C. H. Hoi. Learning cross-modal embeddings with adversarial networks for cooking recipes and food images. In *CVPR*, 2019.

[Wang et al., 2019b] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Peng Cui, Philip S. Yu, and Yanfang Ye. Heterogeneous graph attention network. In *WWW*, 2019.

[Xie et al., 2020] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *CVPR*, 2020.

[Xu et al., 2019] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[Zhang et al., 2019a] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In *KDD*, 2019.

[Zhang et al., 2019b] Chuxu Zhang, Ananthram Swami, and Nitesh V. Chawla. Shne: Representation learning for semantic-associated heterogeneous networks. In *WSDM*, 2019.