Recipe Prediction and Completion with Attentioned Multipartite Network

Daeho Lee
Graduate School of BBE, KAIST
Daejeon, South Korea
daeho715@kaist.ac.kr

Sang Un Gu
Kim Jaechul Graduate School of AI, KAIST
Daejeon, South Korea
ayslaon311@kaist.ac.kr

Youngjun Im
Kim Jaechul Graduate School of AI, KAIST
Daejeon, South Korea
youngjun@kaist.ac.kr

ABSTRACT

This project is about prediction and completion task of given 35317 recipes, 6714 ingredients and 20 cuisines. To handle these tasks, we decided to use recommendation system. Recently, with applying neural network methods into graph structures, especially graph convolutional networks(GCN) give the best performance in the recommendation tasks. However, many graph neural networks were lack of scalability due to the ineffectiveness of Laplacian matrix. To handle this issue, works such as PinSAGE and MultiSAGE based on random-walk have been done.

In our task, there is need to consider connectivity of ingredients and cuisines represented as recipes. Therefore, we need to consider the graph which can understand the contexts; recipes. So we decided to construct heterogeneous graph consisting of three types of nodes of ingredients, recipes and cuisines rather than bi-partite graph in PinSAGE. We'll follow the contextualized multi embeddings in MultiSAGE expected to be able to consider importance of the weights in connected nodes.

1 INTRODUCTION

Predicting cuisine from ingredients is to solve the completion task and classification task by using 6714 ingredients, 35317 recipes, and 20 cuisines. The completion task is to find a missing ingredient in a given recipe. The classification task is to find cuisine using a perfect recipe. The completion task predicts missing ingredients, so it can be helpful for grocery shopping or cooking at home. The classification task predicts the cuisine for each recipe, so it can know ingredients that are frequently used in a particular cuisine. Therefore, it can be helpful in trade or exchange by each country.

We decided to use a recommendation system to solve these problems. A *recommendation system* is a tool that assists users by presenting services or products that are most likely of their interest. In general, recommendations can be generated based on user preferences, item features, user-item transactions, and environmental factors such as time, season, and location[3].

Recently, the best performance of the recommendation system is the graph neural network, one of the deep learning architectures. Among graph neural networks, convolutional neural networks (GCNS) have received significant attention for the task of extracting information from large graphs[2].

GraphSAGE (SAmple and agreGate)[4] was created for inductive learning of GCNS. GraphSAGE trains a set of aggregator functions that learn to aggregate feature information from a node's local neighborhood. Among the recommendation system based on GraphSAGE are PinSAGE and MultiSAGE.

PinSAGE[7] is a random-walk-based GCN. In PinSAGE, Pin and Board are expressed in a bipartite graph. The Pin is a general item, and the board represents multiple pins. Personalized Pagerank algorithm (PPR)[1] can be utilized as high-order heuristics that consider neighbors of the target node in the whole network for calculating the similarity score. The aggregate in PinSAGE used a PPR. PinSAGE captures connection information between pins through Metapath(Pin-Board-Pin) and embedding only for pins. Thus, PinSAGE considers the graph to be a homogeneous network of the Pin. Since the information on the board is used only as a metapath, there is a disadvantage in embedding information of node types other than the Pin cannot be generated.

PPR is a pooling score that does not reflect the feature information of nodes. Thus, MultiSAGE[6] introduces the concept of graph attention network(GAT)[5]. Also, unlike PinSAGE, which created and learned graphs in the form of bi-partite, MultiSAGE enables contextualized multi embeddings. That is, The embedding type can be different according to the situation.

If a graph is made using the connectivity of ingredients, recipes, and cuisines in the predicting cuisine problem, it becomes a heterogeneous network with three types of nodes. Therefore, if each node is practiced as an essential weight of a connected node through node embedding of MultiSAGE, it will perform well in the completion task and classification task.

2 PROPOSED METHOD

2.1 Preliminaries

The three types of attributes exist in the graph.

- Node attributes (e.g., node identity, number of neighbors)
- Edge attributes (e.g., edge identity, edge weights)
- Global attributes (e.g., number of nodes, longest path)

The objective of the project is to complete recipes and classify cuisine aligns with learning attributes of nodes, especially those of ingredients. The provided dataset possesses an apparent heterogeneity as visualized in Figure 1. In the graph's global scope, ingredient nodes can be considered target nodes, while recipe nodes and cuisine nodes can be considered context nodes.

2.2 MultiSAGE

Our method includes the idea from MultiSage, a graph convolutional network for a recommender system. The MultiSAGE[**?**] is a GCN engine with contextual information, modeling interactions between target nodes and context nodes in the multipartite network. The key improvements of MultiSAGE from PinSAGE are *contextual masking* and *contextual attention.*

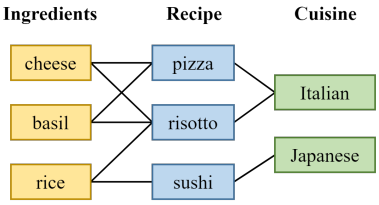

Figure 1: Simplified example of constructed graph.

Contextual masking aggregates the target embeddings $\mathbf{z}_t$ based on context embeddings $\mathbf{z}_c$ as following equation.

$$\mathbf{z}_{t|c} = \mathbf{z}_t \otimes \mathbf{z}_c \tag{1}$$

Since the last embedding layer of $\mathbf{z}_c$ is the ReLU activation function, which sets dimensions with negative values as zeros, the target embeddings' irrelevant dimension is masked out by the corresponding context. This relational operation has intuitive geometrical meanings, and has shown empirical predominancy over popular operations such as summation $\mathbf{z}_t \oplus \mathbf{z}_c$ and dense neural network $\text{RELU}(\mathbf{W}_p(\mathbf{z}_t \otimes \mathbf{z}_c) + \mathbf{b}_p)$.

Contextual attention accounts for the context-wise dynamic impact of neighbor nodes with regard to the ego node. The attention of edge is jointly computed by the ego-context-neighbor relationship as follows.

$$\alpha(v,o,u) = \frac{\exp\left(\tau(\mathbf{a}^T[\mathbf{W}_{at}\mathbf{z}_t(v) \odot \mathbf{W}_{ac}\mathbf{z}_c(o) \odot \mathbf{W}_{at}\mathbf{z}_t(u)])\right)}{\sum_{\substack{u' \in \mathcal{N}_v, \\ o' \sim (v,u')}} \exp\left(\tau(\mathbf{a}^T[\mathbf{W}_{at}\mathbf{z}_t(v) \odot \mathbf{W}_{ac}\mathbf{z}_c(o') \odot \mathbf{W}_{at}\mathbf{z}_t(u')])\right)} \tag{2}$$

Here, $\mathbf{W}_{at}$ and $\mathbf{W}_{ac}$ are the learnable parameters for target embedding and context embedding, respectively. The multi-head attention mechanism is used to aggregate contextualized embedding as follows:

$$\mathbf{z}_{\mathcal{N}_v}(x) = \sigma\left(\frac{1}{D}\sum_{d=1}^{D}\sum_{u \in \mathcal{N}_v, o \sim (v,u)} \alpha^{(d)}(v,o,u)\mathbf{z}_{t|c}(x,o)\right), \tag{3}$$

where σ is a sigmoid function.

The training objective is a max-margin ranking described as follows,

$$\mathcal{J}(v_q, v_p, v_n) = \max\{0, \mathbf{h}_{v_q}^T \mathbf{h}_{v_n} - \mathbf{h}_{v_q}^T \mathbf{h}_{v_p} + \delta\}, \tag{4}$$

where δ is a margin hyper-parameter. The v_q, v_p, v_n corresponds to the query node, positive node and negative node, respectively, sampled from the training data.

The advances of this project from the original MultiSAGE algorithm are three-fold.

- We empirically demonstrate the claim in [6] that MultiSAGE can incorporate multiple context types simultaneously during the training and testing phase.
- Furthermore, we extend the *level* of context nodes and show that indirect contexts are also effective for learning target features. For example, as depicted in Figure 1, we utilize indirect context *cuisine* that are not directly connected to target node *ingredients*.

$\lvert V\rvert$	30281	$\lvert V_I\rvert$	6714
		$\lvert V_R\rvert$	23547
		$\lvert V_C\rvert$	20
$\lvert E\rvert$	277006	$\lvert E_{IR}\rvert$	253459
		$\lvert E_{RC}\rvert$	23547

Table 1: Some statistics of graph. $|V|$ and $|E|$ are number of nodes and edges in the graph, respectively. $|V_I|, |V_R|, |V_C|$ are number of nodes with type *ingredient, recipe, cuisine*, respectively. $|E_{IR}|, |E_{RC}|$ are edges between ingredient and recipe, recipe and cuisine, respectively.

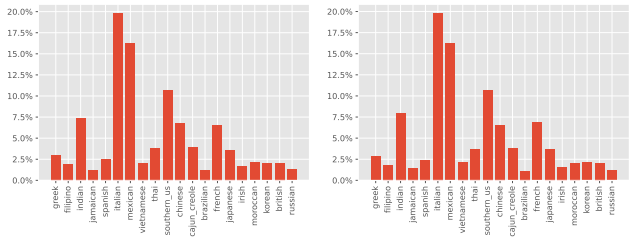

Figure 2: Class distribution of the training set (left) and validation set (right).

- We extend the architecture to perform collaborative filtering and classification over the learned node embeddings.

2.3 Graph construction

Alongside the structure of the graph, the method for the generation of initial node features is an interesting topic. Since the name of every ingredient is given, one might use the NLP-based algorithms to extract feature-based embeddings. However, because given names are relatively short and the graph consists of a relatively small number of nodes, using identity-based embedding is practical and simplifies the network.

3 EXPERIMENTS

We conduct experiments on given dataset which consists of 35317 recipes, 6714 ingredients and 20 cuisines. In following sections, we would compare our results and some results from basic models such as Random Forest, Linear Regression, and Linear SVC.

3.1 Dataset

Dataset consists of 7 files. `train.csv` has 35317 rows where each row represents the recipe. `validation_{classification/completion}_{answer/question}.csv` has 7847 rows for each tasks. `test_{classification/completion}_question.csv` has 3923 rows for each tasks and the test set has no answers.

We're aiming to train the MultiSAGE model suggested to solve the given tasks with `train.csv` and validate the performance of model with `validation*.csv` files and submit the answers of the test set.

The class distributions of the train and validation set are shown in figure Figure 2, respectively.

Methods	Micro F1-score	Macro F1-score	Accuracy
LinearSVC	0.635	0.510	0.635
RandomForest	0.471	0.210	0.472
Linear Regression	0.638	0.531	0.638
MultiSAGE			

Table 2: Performance measured by {Micro/Macro} F1-score and accuracy of each model in validation classification set.

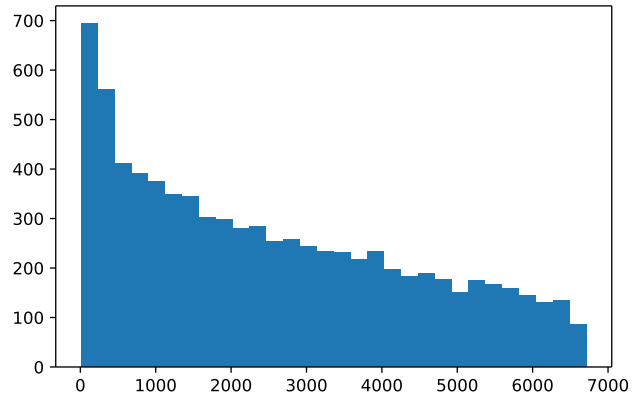

Figure 3: The rank distribution of the correct ingredient in the model's prediction.

3.2 Recipe Classification Task Into Cuisine

The results of classification on validation set using our model and other baseline models are shown in Table 2.

3.3 Ingredient Completion Task

For the completion task, we computed the sum of pairwise score between the ingredients already in the recipe R and the new ingredient i. The missing ingredient can be found by choosing the ingredient that maximizes the pairwise score.

$$Score(R, i) = \sum_{j \in R} \langle h_i, h_j \rangle + b_i + b_j. \tag{5}$$

Here, h is a node embedding and b is a bias term learned by model's scoring function. See `completion.ipynb` for implementation details. The performance of the model for ingredient completion task is shown in Figure Figure 3.

4 CONCLUSIONS

REFERENCES

[1] Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. Fast incremental and personalized pagerank. *arXiv preprint arXiv:1006.2880*, 2010.
[2] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.
[3] Aminu Da'u and Naomie Salim. Recommendation system based on deep learning methods: a systematic review and new directions. *Artificial Intelligence Review*, 53(4):2709–2748, 2020.
[4] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.
[5] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
[6] Carl Yang, Aditya Pal, Andrew Zhai, Nikil Pancha, Jiawei Han, Charles Rosenberg, and Jure Leskovec. Multisage: Empowering gcn with contextualized multi-embeddings on web-scale multipartite networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2434–2443, 2020.
[7] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.

A APPENDIX

A.1 Labor Division

The team will perform following tasks. The division of the labor and the tasks is subject to change.

- Implementation of the multipartite graph [all]
- Implementation of inference and training algorithm [all]
- Experiments on the validation data [all]

A.2 Full disclosure wrt dissertations/projects

Daeho Lee: He is not doing any dissertation or project related to this project: he is in area of Bioinfomatics (drug discovery using AI).

Youngjun Im: He is not doing any dissertation or project related to this project: his research is on modeling point process using graph recurrent neural network.

Sang Un Gu: He is not doing any dissertation or project related to this project: he is in the area of explainable AI.