GraphShop: Graph-based Approach for Shop-type Recommendation

Guoyuan An* Sungeui Yoon*† Jaeyoon Kim* Lin Wang* Myoungho Kim*†

Abstract

It is essential to predict the popularity of a particular shop type when investors decide which type of shops to open at a given location. Existing shop-type recommender systems have approached this problem by building a region-type matrix and analyzing the relationship between different regions and shop types. However, these methods make recommendations for each region, thus having difficulty analyzing a specific shop, especially near the two regions' borders.

To tackle this challenge, we propose a novel Graph Neural Network (GNN) model, called GraphShop, to represent shops as nodes in a graph and analyze each shop without assigning it to a region. As it is difficult to find the influential neighbors, we propose two aggregation methods, DistanceModule and TypeModule, in GraphShop. DistanceModule aggregates unordered nearby shops in every zone and filters them from the remote zones. TypeModule reorders the nearby shops based on their types and considers the interaction of different types. Furthermore, to address the lack of open shop-type recommendation datasets, we build a qualitative and large-scale dataset collected from a review website and location-based services. It contains most, if not all, shops in a region and is large and diverse, by containing 53,182 shops with 122 types. Our dataset is available at https://github.com/BoSamothrace/GraphShop. Through the experimental results, we demonstrate that our method outperforms the existing state-of-the-art methods for shop-type recommendation by a factor of up to 37 %.

Keywords— shop-type recommendation, graph neural network, smart city, recommender system

1 Introduction

Cities are growing, and the world's urban population is projected to reach 58% in 2050 by adding 2.5 billion people [8]. Metropolises provide us sustainable and resilient environment as well as business opportunities. At the same time, exabytes of data are created every day. A new way of mining urban-data can bring us great business success. One of the examples is the shop-type recommendation.

Similar to the conventional recommender systems seeking to predict the 'rating' or 'preference' a user would give to an item [12, 28, 32], shop-type recommendation aims to predict the popularity of a shop type at a given space [29]. In real life, different locations might suit different shop types [17], e.g., places near the subway stations are better for the fast-food shops, and places near high-grade districts are more suitable for well-equipped steakhouses. When a vendor gets an available location and needs to choose one shop type to start a new business, it is imperative to predict each type's popularity, which will help estimate future profit and reduce investment risks.

To achieve this goal, business analytic teams usually provide suggestions based on their subjective judgments and perception. With the development of new data mining methods (e.g., collaborative filtering and deep learning) and the rapid growth of data on every aspect of everyday city life, we now have more advanced ways to solve this problem. Yu *et al.* [29] and Mao *et al.* [16] creatively applied collaborative filtering [21, 32] to the shop-type recommendation problem. They divided a city into many regions and built the region-type matrix reflecting how popular a type is for a region. By analyzing the relationship between different regions and shop types using collaborative filtering techniques like matrix factorization [14], they studied this problem at the quantitative level for the first time.

While building the region-type matrix shows benefits for the shop-type recommendation, it faces two challenges: how to deal with shops near the border of two regions and how to distinguish the characteristics of different locations in the same region. As shown in Fig. 1, an available place is assigned to region A, but it is closer to the coffee shops in region B, which might be more influential. Hence, we need to not only consider the region where a shop belongs, but also care about its distances from neighboring shops.

GNN has achieved excellent performance on many recommendation problems [5, 10, 25, 28]. By representing shops as nodes in a graph and analyzing each shop without assigning it to a region, GNN models can be applied to the shop-type recommendations. Nonetheless, we found two issues of naively applying GNN models. First, it is not straightforward to find the meaningful edges (or hyperedges) between shops. Although we could artificially connect nearby shops, these geographically defined edges are not necessarily relevant to the shops' interaction effect in terms of popularity. For example, a famous but far restaurant might influence the target shop a lot, but other shops located in the same distance may not have much impact. Second, a location-based graph is insufficient to analyze the interaction effect between different types. For instance, a steakhouse gives more influence to an English pub than a grocery shop.

We propose a novel GNN framework, called **GraphShop**, for the shop-type recommendation. Regarding the challenge of mismatch between geographical edges and popularity relationships, we propose **DistanceModule** to *filter* out shops' influence from ones located far away so that only vital information remains. As shops' locations are unordered, DistanceModule learns an order-robust function to aggregate the shops' information

*KAIST, {anguoyuan,sungeui,sgrbq,wanglin,mhkim}@kaist.ac.kr
†Supported by KISTI/KAIST joint research fund

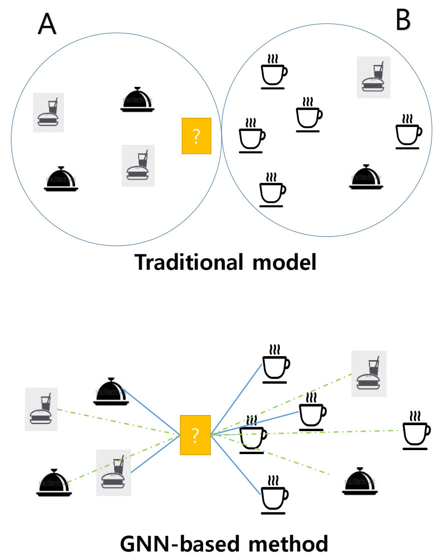

Figure 1: A prediction case belongs to region A, but gets more influence from region B. Different with previous methods based on region-type matrix [16, 29], GNN-based model does a specific analysis for each shop and considers the distance between them. The yellow boxes are the available places, blue lines are short-distance edges and green dotted lines are long-distance edges.

in a zone before the filtering process. To tackle the complex interactions between different types, we propose **TypeModule**, which *sorts and hierarchically aggregates* similar types' information. In such a way, GraphShop can model a city as a graph and analyze each shop separately by aggregating its neighborhood information and balancing the influence of features, distances, and types.

Another critical issue in the shop-type recommendation is the lack of an open dataset, hindering active research on this problem. In this paper, we build the first public available shop-type recommendation dataset. Compared with the unopened datasets collected by [16, 29], our new dataset has three benefits: 1) Concentration: our dataset contains most, if not all, shops in one district. 2) More shops: a total of 53,182 shops' information. 3) More shop types: contains 122 types. We provide the details in Sec. 4.

To validate the effectiveness of GraphShop, we compare with the state-of-the-art shop-type recommendation methods, e.g., feature fusion matrix factorization (FFMF) [16, 29]. We also compare ours with two multi-layer perceptrons (MLP) based methods commonly used in other recommendation problems: multi-layer perceptron (MLP) and neural matrix factorization (NeuMF) in [12]. To show the superiority of GraphShop over the existing GNN models in other recommendation problems, we connect the nearby shops and apply GraphSage [10], one of the most famous GNN models to predict the popularity. Note that we are the first to apply NeuMF and GNN models on the shop-type recommendation problem. Experimental results show that GraphShop outperforms FFMF, MLP-based method, and GNN baseline by 40%, 37%, and 16%, respectively.

Overall, this paper's main contributions are three-fold:

- We propose GraphShop, to the best of our knowledge, the first GNN-based method for the shop-type recommendation. It represents shops as nodes in a graph and analyzes each shop without assigning it in a region.
- We propose DistanceModule and TypeModule in GraphShop to filter and aggregate the influential neighbors. They vastly expand the aggregation range and learn the elaborate influence relations between different node types. The experiment result shows that GraphShop significantly outperforms the existing methods.
- We build and open a large real-world dataset for the shop-type recommendation, collected from the social media and location-based services for broader use by the research community.

2 Related Work

Shop-type Recommendation Designing recommender systems has been an effective strategy to offer proper matches for massive entities in our daily life. The most common utilization is matching users and items based on users' preferences. Companies and scientists could not only recommend movies and articles, but also restaurants [3, 7], travel plans [30], medical treatment [23] and so on to users.

In addition to the relations between users and items, recommender systems are powerful enough to exploit the connections between locations and shops. Because of the high commercial value, site selection for a new business facility has been popular for a long time [13, 31]. Fu [6] exploited the geographic dependencies in real estate, Karamshuk *et al.* [13] proposed a method to identify the optimal location for a new retail store by considering the popularity and people's mobility of places. Yu *et al.* [31] investigated the effectiveness of user-generated content in the location-based social network for selecting the shop's location. These approaches focused on choosing the best location for a fixed store type.

On the other hand, another critical scenario - an investor has an available shop space but does not know the popularity of each shop type - has emerged recently. The number of reviews could measure the popularity of an existing shop. He *et al.* [11] and Wu *et al.* [26] have showed the high correlation between the number of reviews and popularity. Some studies took the number of reviews as a popularity indicator [4, 19]. The difficulty is how to find the relation between different places and different shop types. Yu *et al.* [29] and Mao *et al.* [16] have done pioneering works on this problem. Yu *et al.* [29] used collaborative filtering to recommend shop types for a given location. Mao *et al.* [16] introduced more features (e.g., user's ratings for each shop) to improve the performance. These methods require the region-type matrix, which divides a district into several regions to compare the distribution of types of those regions (Sec. 5).

Different from the approaches mentioned above, GraphShop analyzes every available location separately by representing it as a node in a graph. Instead of item rating (item ranking and rating prediction are the two most common tasks in recommender systems [9]), we evaluate the performance of popularity prediction, which is more cared for by the shop investors. By using the topology structure of the graph, our method outperforms the former approaches significantly.

GNN Models Graph models have achieved excellent performance on many data mining applications, and the critical mechanism is aggregating the information of neighbor nodes through the edges [32]. Most of the applications of GNN have clear relationships between the vertexes (e.g., users evaluate products, researchers cite the works of other researchers). Other applications use heuristic edges (e.g., the distances of sensors in the traffic prediction problem [27]) and auxiliary information (e.g., temporal changes of the nodes [18, 27]) for GNN, to update the embedding vector of each node [15]. However, the shop-type recommendation problem has neither transparent edges (we do not know how each shop will reflect another shop exactly) nor temporal data. Although there are more than 50,000 shops in our dataset, no two shops share the same neighborhood information. Thus the difficulty comes to how to filter real influential edges among all the geographically defined edges. GraphShop tackles this challenge by using DistanceModule to aggregate information from neighbors at different distances and using TypeModule to consider different types' influence.

Other Related Applications Another two similar applications are semantic segmentation and graph similarity. We know how pixels (and points) are located near each other in images and points cloud, but do not know their relations. Pixels in images arrange as Euclidean geometry, thus could benefit from the CNN architectures. The CNN channels with large receptive fields correspond to our update gate technique, which aggregates the information from far area; channels with small receptive fields correspond to our nearby context and type context, which aggregate the nearby neighbors' information. Although point clouds are non-Euclidean, the segmentation for point clouds (and images) relies on upsampling [20], which is different from GraphShop that analyzes each specific target shop and target type. We could also regard shop-type recommendation as a sub-graph similarity problem. However, the traditional graph similarity [2] only focuses on the node features and graph structure, where the sub-graph in the shop-type recommendation is radial gradient and focuses more on how the sub-graph effects the target type.

3 The Proposed Method

3.1 Definitions and Notations Let $V = \{v_1, v_2, \dots, v_n\}$ be the set of vertices, where each vertex denotes a shop. Each shop v_i is associated with its location l_{v_i}, shop type t_{v_i}, and some shop features f_{v_i}, such as rating and price. All the shops together constitute a complete graph $G = (V, E)$, where every geographically defined edge $e_{i,j} \in E$ is associated with a distance value $d_{i,j} \in \mathbf{D}$. Here, $\mathbf{D} \in \mathbb{R}^{n \times n}$ is the distance matrix and $d_{i,j}$ denotes the distance between shops v_i and v_j.

Let $N(i, r_1, r_n) = \{v_j | r_1 < d_{i,j} \le r_n\}$ be the set of v_i's neighbor shops whose distance to the shop v_i is larger than r_1 and small than or equal to r_n. T= {fast food shop, coffee shop, $\cdots$, korean food shop} is the set of all the possible types. We use an embedding vector $\mathbf{q}_{t_k} \in \mathbb{R}^d$ to denote a shop's type $t_k \in T$. Given a shop's location and type, we want to predict its popularity by analyzing the types, prices and popularities of its neighbor shops. The mathematical notations used in this paper are summarized in Table 1.

Symbols	Definitions and Descriptions
V, v_i	the set of vertices, and a i-th vertex in V
E, $e_{i,j}$	the set of edges, and an edge between vertices v_i and v_j.
$\mathbf{D}$, $d_{i,j}$	distance matrix, and the distance between vertices v_i and v_j
$N(i, r_1, r_2)$	the set of v_i's neighbor shops
T, t_k	the set of all possible types, and a type in T
t_{v_j}	the type of vertex v_j
$\mathbf{q}_{t_k}$	the embedding of type t_k
$\mathbf{h}_i^F$	the contextual hidden state of far area
$\mathbf{h}_i^N$	the contextual hidden state of near area
$\mathbf{h}_i^T$	the contextual hidden state of similar types
$\hat{p}_{i,k}$, $p_{i,k}$	the predicted and true popularity of type k on vertex v_i.

Table 1: Notations table

3.2 DistanceModule The first challenge is that we do not know which neighbor shop influences the target shop. As mentioned in Section 1, the geographically defined edges are not necessarily relevant to the shops' interaction effect in terms of popularity. Some shops could influence the shops far from them, and some only influence the shops just close to them. As we do not know each shop's effective radius, we propose the DistanceModule to aggregate neighbors' information from a very far distance and analyze their relations; e.g., if one shop has no impact on shops 0.5km far away, it has a low probability of influencing the shops locating 1km away. Indeed, this process deals with the connections between nodes sequentially, as the Recurrent Neural Network (RNN) does. We, therefore, design an RNN-based filtering block for learning the patterns.

Also, as the number of one-to-one pair permutations is factorial to the number of neighbors, considering each shop is complex and time-consuming. Instead of studying shop-to-shop relations, we study the **inter**-zone relations and keep the characteristics state of the far zone if it influences shops' characteristics in the near zone. (Note that the concept of zone in GraphShop and region in previous works [16, 29] are different. Zones are defined based on the distance with each

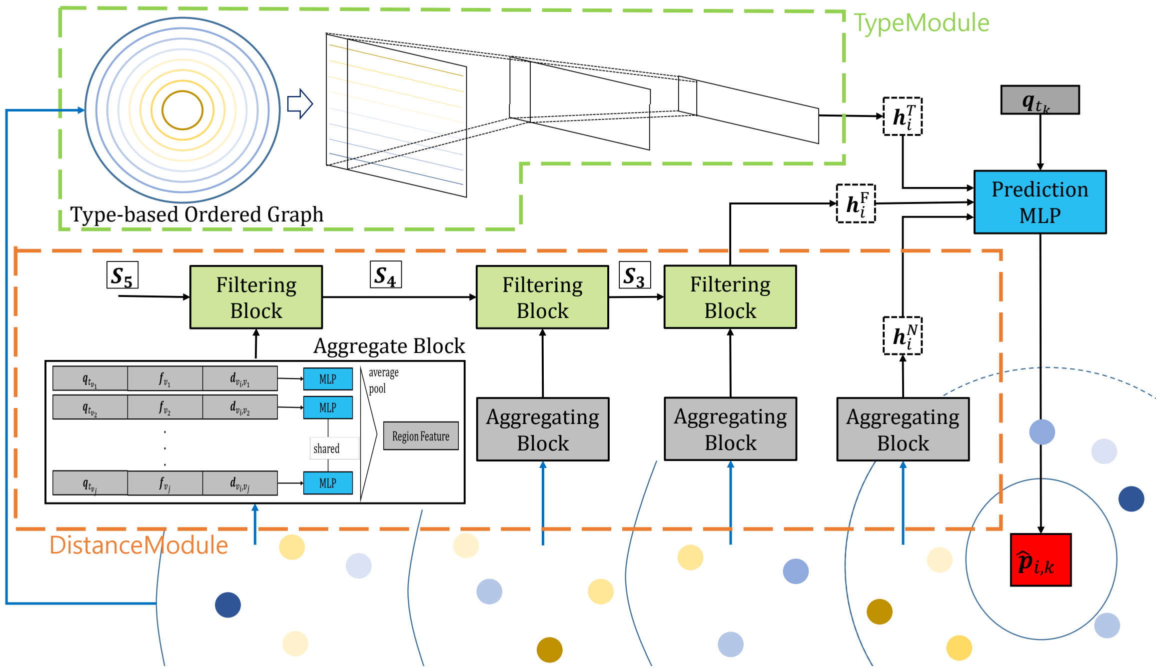

Figure 2: The overview of GraphShop. The final prediction MLP predicts the popularity $\hat{\mathbf{p}}_{i,k}$ based on the type embedding $\mathbf{q}_{t_k}$, the $\mathbf{h}_i^F$ and $\mathbf{h}_i^N$ learned by DistanceModule and the $\mathbf{h}_i^T$ learned by TypeModule. The blue arrows represent the process of inputting the nearby shops' information.

target shop, but regions in [16, 29] are defined by dividing a map.) However, how to deal with **intra**-zone relations becomes a new challenge, considering the shops in a zone are unordered.

Inspired by the work of Qi et al. [20] and the pooling aggregator of GraphSage [10], we design the aggregating block, which uses the MLP function and symmetric function to approximate a continuous set function. [20] has given the mathematical proof that this simple neural network architecture is robust to the perturbation of items in the set. That is, the perturbation in the set will not greatly change the approximated function values. Moreover, an aggregating block represents different distances using embedding vectors to learn how distance cooperates with other shop information, such as price and rating, to affect the target shop's popularity.

Particularly, we divide the set of neighbor shops $N(i, r_1, r_n)$ into $N(i, r_1, r_2)$,$N(i, r_2, r_3)$,...,$N(i, r_{n-1}, r_n)$, where $r_1 < r_2 < ... < r_n$, and aggregating block calculates their characteristic vectors as follows:

$$\begin{aligned} \mathbf{c}_{v_j} &= p[\mathbf{q}_{t_{v_j}} \oplus \mathbf{f}_{v_j} \oplus \mathbf{d}_{v_i, v_j}], \\ \mathbf{o}_{n-1} &= \sum_{v_j \in N\left(i, r_{n-1}, r_n\right)} \alpha_{n-1} \mathbf{c}_{v_j}, \\ &\quad \ldots \quad , \\ \mathbf{o}_1 &= \sum_{v_j \in N(i, r_1, r_2)} \alpha_1 \mathbf{c}_{v_j}, \end{aligned} \tag{3.1}$$

where $\mathbf{f}_{v_j}$ is the vector of shop features (e.g., ratings, prices), $\mathbf{d}_{v_i, v_j}$ is the embedding vector of distance, p denotes the MLP function in aggregating block, and α_{n-1}, α_{n-2}, ..., α_1 are set to $\frac{1}{\left|N\left(i, r_{n-1}, r_n\right)\right|}, \frac{1}{\left|N\left(i, r_{n-2}, r_{n-1}\right)\right|}, \ldots, \frac{1}{|N(i, r_1, r_2)|}$, respectively. $\mathbf{c}_{v_j}$ is the characteristic vector of shop v_j, $\mathbf{o}_k$ is the characteristic vector of $N(i, r_k, r_{k+1})$.

Then, the output of filtering block is given by:

$$\hat{\mathbf{o}}_k, \mathbf{s}_k = f(\mathbf{o}_{k+1}, \mathbf{s}_{k+1} | \theta), \tag{3.2}$$

where f denotes the RNN loop, $\hat{\mathbf{o}_k}$ and $\mathbf{s}_k$ are the predicted characteristic vector and RNN hidden state at step k. We use $\mathbf{s}_2$ as the contextual hidden state of the far region $\mathbf{h}_i^F$ and $\mathbf{o}_1$ as the contextual hidden state of the near region $\mathbf{h}_i^N$.

3.3 TypeModule In addition to distance, the interaction relation among shop types is another critical factor. When predicting the popularity of the target shop, the information of similar shop types is more valuable than unrelated types. For example, a sandwich shop's information is more helpful than a nuts shop for predicting the popularity of a hamburger shop. To better cope with the interaction relations among shop types, we present TypeModule to give different weights to different types. Given a target type, the neighbors shared similar types should get larger weights. In Figure 2, we use shorter edges to delegate larger weights and call this reordered graph as type-based ordered graph.

There are, however, two challenges when aggregating the information in the type-based ordered graph: a large number of types (total 122 types) and the sparse random sampling of each type (target shop can have many nearby shops, but few for every type). Thus the simple attention technique is time-consuming, and difficult to infer the real influence of each type. To overcome these challenges, we propose to increase the sampling amount by combining the very close types first. That is, do clustering on the shop types and give each type group the same weight. Inspired by using 1-D kernel on sequence data [1, 22], to implement a real-time hierarchical clustering, we apply the CNN layers to aggregate properties of shop types in the type-based ordered graph.

Specifically, we first divide the neighbor shops in the set $N(i, r_1, r_n)$ based on their types. For each type, we get its characteristic vector as follows:

$$\begin{aligned} \mathbf{c}_{v_j} &= [\mathbf{q}_{t_{v_j}} \oplus \mathbf{f}_{v_j} \oplus \mathbf{d}_{v_i,v_j}], \\ \mathbf{c}_{t_k} &= \sum_{v_j \in \{v_j | t_{v_j} = t_k, v_j \in N(i,r_1,r_n)\}} \mathbf{c}_{v_j}, \end{aligned} \tag{3.3}$$

where $\mathbf{c}_{v_j}$ is the characteristic vector of shop v_j and $\mathbf{c}_{t_k}$ is the characteristic vector of a specific type $t_k \in T$.

GraphShop uses embedding vectors to represent types, and the similar types will share close embedding vectors in the latent space through training. In this step, GraphShop sets the target shop type as the base shop type, then sort and concatenate the other shop types' characteristic vectors in an ascending order based on the Euclidean distances between their embedding vectors and the target shop type's embedding vector. We finally get the type-based matrix $\mathbf{M}_i$ for the shop v_i; each row of $\mathbf{M}_i$ corresponds to characteristics vector of each shop type.

Thanks to its construction method, a local region across different rows in this type-based matrix are related to each other. We transfer the type-based matrix $\mathbf{M}_i$ into a tensor $\boldsymbol{M}_i^0 \in \mathbb{R}^{a_0 \times b_0 \times 1}$ (a_o and b_o are the number of types and the the number of features respectively.) and then feed it to a number of CNN layers.

Assuming that $\boldsymbol{M}_i^{l-1}$ denotes the feature maps in the $(l-1)$-th layer, the output of the l-th layer is given by:

$$\boldsymbol{M}_i^l = f\left(\mathbf{W}^l * \boldsymbol{M}_i^{l-1} + b^l\right), \tag{3.4}$$

where $*$ denotes the convolutional operation, $f(\cdot)$ is the activation function, $\mathbf{W}^l \in \mathbb{R}^{k_l \times k_l \times d_{l-1} \times d_l}$ denotes d_l (the number of feature maps in l-th layer) convolutional kernels of size $k_l \times k_l \times d_{l-1}$, $b_l \in R^{d_l}$is a bias term, and $M_i^l \in R^{a_l \times b_l \times d_l}$ denotes the output feature maps at l-th layer.

We flatten the final feature map M_i^l as the contextual hidden state of similar types $\mathbf{h}_i^T$ and use it to predict the popularity of target shop with contextual hidden states of the far and near areas as following:

$$\begin{aligned} \mathbf{h}_i^G &= p\left([\mathbf{h}_i^F \oplus \mathbf{h}_i^N \oplus \mathbf{h}_i^T]\right), \\ \hat{p}_{i,k} &= q\left([\mathbf{h}_i^G \oplus \mathbf{q}_{t_k}]\right), \end{aligned} \tag{3.5}$$

where $\hat{p}_{i,k}$ is the predicted popularity for shop v_i and type t_k, $\mathbf{h}_i^G$ is the global contextual hidden state of shop v_i, and p and q denote the MLP functions.

4 Open A Large-scale Dataset

Yu *et al.*and Mao *et al.* [16, 29] collected datasets from review websites and map applications, but they are not publicly available, thus making it impossible to train and evaluate GraphShop. To overcome this critical issue, we collect and build a new dataset containing abundant information about the shop types and their locations. To the best of our knowledge, we are the *first* to build and release a dataset publicly available for broader use by the research community for the shop-type recommendation.

Our dataset follows the protocols in Yu *et al.* [29] and Mao *et al.* [16]. The preparation of the dataset follows several steps. Firstly, we crawled shop-related data in Beijing city from 'Dianping.com', which is one of the largest review websites in China and provides credible information about shops. Secondly, we obtained the latitude-longitude address using the Baidu Map's place API and make the distance matrix $\mathbf{D}$, as explained in Sec. 3.1. Our dataset includes name, type, grade, the number of comments, price, service score, environment score, flavor score, acceptance of group order, region, and address of each shop. We totally collected four sub-datasets from 5 different districts in Beijing (notice that A/B-urban includes two districts, Docheng and Xicheng districts). The detailed information of the datasets is summarized in Table 2, two cases are demonstrated in Table 3, and the geographic positions of the selected districts are shown in Figure 3.

Compared with existing shop datasets, our dataset has the following benefits: 1) Concentration. All the 53182 shops in our dataset are located in five districts in Beijing (Beijing has 16 districts in total). Compared with Yelp dataset, which doesn't guarantee the proportion of total shops in a metropolitan, our dataset contains most, if not all, shops in one district. Hence, this dataset better reflects the neighborhood information of every shop and is more proper to be used for shop-type recommendation. 2) A more massive amount of data. Our dataset incorporates the information about 53182 shops in total, while Yu *et al.* [29]'s dataset containing 17435 shops and Mao *et al.* [16] collected the dataset of 29763 shops. 3) More variety of shop types. Our dataset includes 122 shop types, while Yu *et al.* [29] and Mao *et al.* [16] tested the recommender system's performance less than ten shop types. We check the performance of each method using our dataset containing four sub-datasets. In the experiments, 60% of the dataset is used as the train set, 20% as the validation set, and the remained 20% as the test set.

5 Experiments

5.1 Experimental Setup

Implementation Details We use Python 3.6 and Pytorch 1.4.0 to implement all the methods. In GraphShop, the type embedding size is set to 16, and MLP layers in Eq. 3.5 are set as [48,64,64,16] and [32,16,8,1] respectively. We use two convolutional layers with kernel size 3 and stride 1 in eq. 3.4 and use 3 LSTM layers with hidden size 16 in eq. 3.2. We optimize the model with mini-batch Adam,

Dataset	Shop numbers	District	Area
Urban	9808	Dongcheng and Xicheng district	$95.56km^2$
Haidian	11923	Haidian district	$431km^2$
Chaoyang	23410	Chaoyang district	$475km^2$
Fengtai	8041	Fengtai district	$306km^2$

Table 2: Dataset summary.

ID	***7	***5
Type	Guangdong tea restaurant	Roast fish
Comments number	15	175
Average price	82	114
General rating	3+	4+
Service rating	7.1	8.7
Environment rating	7.7	8.7
Flavor rating	7.6	8.5
Region	Chaoyang Sanlitun	Chaoyang Guomao
Coordinate	***	***

Table 3: Two cases in our dataset

Figure 3: 16 districts in Beijing. 1 Dongcheng district; 2 Xicheng district; 3 Chaoyang district; 4 Haidian district; 5 Fengtai district; 6 Shijingshan district. Ref: ebeijing.gov.cn

and tested the batch size and learning rate of [10,20,32] and [0.001,0.003,0.01,0.03]. The embedding size of type and location in all the compared algorithms is also 16. The hidden layers in the MLP of the compared algorithms (including method MLP, FFMLP as well as neural CF layers in NeuMF and FFNeuMF) are set as 3. We first train GraphShop with DistanceModule and TypeModule separately, and then use these pre-trained weights [12] to implement our final model.

Evaluation Metric Item ranking and rating prediction are the two most common tasks in recommender systems [9]. In this paper, we focus on the task of rating prediction and use the logarithmic number of comments as the ground truth value, since we care the scale instead of the absolute value of the comments number. For example, the popularity difference between 1 comments and 101 comments is much more than the difference between 800 comments and 900 comments. This scaled outputs also prevent the exploding gradients in the training process. We use Mean Square Error (MSE) to measure the closeness of the predicted popularities to their ground truth values. MSE is the most common metric for rating prediction tasks in recommender systems [24]. Notice that small improvement in the MSE term can reflect a significant enhancement of the accuracy of predicted comment numbers.

5.2 Compared Algorithms We compare our GraphShop with the state-of-the-art shop-type recommendation methods, FFMF. To show an extensive evaluation result, we also compare GraphShop with the deep-learning-based methods and GNN models commonly used in other recommender systems.

Matrix Factorization (MF) Matrix factorization is the most popular method for recommendation, and it has been proven useful for the shop-type recommendation problem [29].

Feature Fusion Matrix Factorization (FFMF): Yu *et al.* [29] and Mao *et al.* [16] have proved that integrating extent features (location features and commercial features) into the basic collaborative filtering is useful for improving the performance [29]. We follow their process and fuse the extent features into the MF method. Particularly, the extent features are set as shop diversity, competitiveness, average number, price, rating, and the number of reviews of all the neighbor shops during the experiment in this paper.

Multi-Layer Perceptron (MLP): This method concatenates the embeddings of region and type, and uses an MLP function to predict the final result [12].

Feature Fusion Multi-Layer Perceptron (FFMLP): FFMLP integrates location features and commercial features into the basic MLP method. The details of feature information are the same as FFMF.

Neural Matrix factorization (NeuMF): This is a factorization model with a neural network architecture [12]. It combines GMF (generalized matrix factorization) with MLP, and has achieved excellent performances in many recommendation problems.

Feature Fusion Neural Matrix factorization (FFNeuMF): it integrates location features and commercial features into the basic NeuMF method. The details of feature information are the same as FFMF.

GNN baseline-mean and GNN baseline-pooling [10]: To show the effectiveness of our proposed model, we artificially connect the target shop with its nearby shops within $300m$, the best radius according to [16]), and use the famous GraphSage [10] as the base GNN method, which has shown the state-of-the-art performance in many other applications. [10] has presented three aggregators to ensure that GraphSage can be trained and applied to arbitrarily ordered node neighborhood feature sets: mean aggregator, LSTM aggregator, and pooling aggregator. As LSTM aggregator shows similar accuracy with pooling aggregator but is significantly slower in [10], we only test mean aggregator and pooling aggregator, denoted as GNN baseline-mean and GNN baseline-pooling, respectively. To keep fairness, we keep other parts of the base GNN same as GraphShop, such as the total number of neighbors, the embedding size, and MLP functions in the prediction process. The only differences between the base GNN models and GraphShop are the DistanceModule and TypeModule, as shown in Section 3.

5.3 Performance Result As the shop-type recommendation does not have real-time requirement (no requirement about speed and complexity), we consider the accuracy as the only performance factor. We summarized the overall popularity prediction error w.r.t. MSE among all the methods on eight datasets in Table 4. We also show some case studies in the Appendix.

GraphShop achieves the ***best*** performance across all the datasets, significantly outperforming the existing methods. On average, GraphShop outperforms the previous state-of-the-art method FFMF by 40% and the GNN baseline by 16%, respectively.

GNN-based methods significantly outperforms the non deep-learning-based methods and MLP-based methods. This shows that our proposed approach of presenting each shop as a node in the graph is effective. Although the prediction part and all the settings in GNN baselines and GraphShop are the same, GraphShop significantly outperforms GNN baselines that we prepared. There is no significant difference between the two GNN baselines. This result shows that our proposed techniques of aggregating from the very far area and considering the relations among different types in GraphShop are useful. We show the detailed effect of these two techniques in the ablation study part.

Among the methods (MF, MLP, and NeuMF) without extent features (Sec 5.2), MLP achieves the best performance, followed by NeuMF and MF. MLP reduces the error of MF by 62% and outperforms NeuMF by 20% on average. Among the methods (FFMF, FFMLP, FFNeuMF) with extent features, FFMF [16, 29] has the best performance. On average, FFMF outperforms FFMLP and FFNeuMF by 13% and 5.7%, respectively.

Although extent features (Sec 5.2) increase the performance of MF significantly (FFMF outperforms MF by 60 %), they do not show many positive effects on MLP and NeuMF. One can design more effective network modules to extract more salient features to improve the performance of DL-based approaches, such as MLP and NeuMF; however, we leave this line of research direction as one of our future work.

5.4 Ablation Study We conduct ablation studies to examine the effectiveness of our proposed aggregation techniques, as shown in Table 5.

DistanceModule itself achieves good performance, outperforming the state-of-the-art method FFMF by 32% and the GNN baseline-mean by 4.5%, as shown in Table 4 and Table 5. This shows that our aggregating block and filtering block are useful for aggregating the vital information of neighbors.

The overall performance rises by combining DistanceModule and TypeModule, mainly reflected on the result of Chaoyang; MSE decreases by 33% over single DistanceModule. After DistanceModule aggregates and filters the nearby shops' information, TypeModule gives different weights to different types, thus improving performance. TypeModule shows different performances on the four datasets. We think the main reason is related to the different type distributions. If a target shop has little similar types near it, theoretically, TypeModule will not show much performance. We leave the analysis of the four datasets' type distributions and improving the TypeModule as our future work.

6 Case study

6.1 Case 1: A Malatang shop in research centre

Target type: Malatang is a popular spicy Chinese street food. Customers pick their desired ingredients, and the chef will cook them in a spicy broth. The price is usually calculated based on the weight of the self-picked ingredients.
Available shop space: The available shop space locates in the important education and research center in Beijing. As shown in Figure 4, this shop location is close to China's two most prestigious universities, Tsinghua University, and Peking University. It is also near the technology hub, Zhongguancun, which is often referred to as "China's Silicon Valley." It is reasonable to deduce that the target customers are university students and employees in technology companies. They usually have high requirements for the convenience and flavor of food, and students also demand a fair price.
Nearby shops: We summarized the existing shops near the target shop place in Table 6. This is a busy trading area. Seven fast-food shops, three afternoon tea shops, and eight other types of shops exist in the region of radius $r = 300$ meters around the target shop. Fast food refers to a shop with a strong priority placed on "speed of service." Unlike Western fast food, usually, hamburgers or french fries, fast food in China is set in a meal of noodles or rice. The popularity of most fast food shops in this area ranges from 1.61 to 2.83 (5 to 17 comments). Afternoon tea shops generally provide light refreshments and a pleasant environment for talking and relaxing, and their popularity ranges from 4.99 to 5.44 (147 to 231 comments). We also notice that the popularity of the shops within 50m with the target location is lower than that of farther shops, which could be useful information for GraphShop to predict the

Dataset	MF	FFMF	MLP	FFMLP	NeuMF	FFNeuMF	GNN baseline -mean	GNN baseline -pooling	GraphShop
Category	Non deep-learning-based		MLP based				GNN based		
Origin target	Shop-type recommendation		Other recommender systems						Shop-type recommendation
Urban	11.94	7.30	7.21	8.32	7.90	7.65	4.75	4.75	**4.35**
Haidian	8.22	6.16	6.10	6.68	6.17	6.13	3.87	3.94	**3.63**
Chaoyang	25.62	7.55	7.08	9.75	9.39	8.74	6.00	6.02	**4.49**
Fengtai	22.38	5.98	5.27	6.39	8.52	6.09	4.60	4.70	**3.62**
Total	68.16	26.99	25.66	31.14	31.98	28.61	19.22	19.41	**16.09**

Table 4: Performance (MSE) of different methods on the tested datasets. Graph models show the best performance across all the datasets. Red numbers numbers denote the **best** results. Notice that small improvement in the MSE term can reflect a significant enhancement of the accuracy of predicted comment numbers as we are using the logarithmic number of comments as the ground truth value.

CNN baseline-mean	DistanceModule	TypeModule	Urban	Haidian	Chaoyang	Fengtai
✓			4.75	3.87	6.00	4.60
	✓		4.39	3.65	6.68	3.63
	✓	✓	**4.35**	**3.63**	**4.49**	**3.62**

Table 5: The result of ablation study about the effectiveness of different components of GraphShop.

Figure 4: The available space for a Malatang shop. The tableware icon refers to the Malangtang shop, and the house icons refer to Tsinghua University, Peking University and Zhongguancun respectively. Our method predicts that the Malatang shop can have 3.59 (36 comments); its ground truth is 3.22 (25 comments). See its nearby shops in Table 6.

final popularity.

Result: GraphShop's prediction result for the Malatang shop is 3.59 (36 comments), while the ground truth is 3.22 (25 comments). Even though all the shops within 50m share comments lower than 17, our GraphShop correctly predicts a Malatang shop might get more attention from customers but not as popular as some farther shops, whose popularity scores are often more than 5 (148 comments).

7 Conclusion and Limitations

This paper aims to tackle two critical issues in the shop-type recommendation: the lack of a publicly available dataset and a generalized Graph-based approach. We first described the proposed method, called GraphShop, to predict a shop type's popularity at a given location. We proposed DistanceModule and TypeModule to filter and aggregate the influential neighbors. We also open a large dataset collected from a review website and location-based services containing 53182 shops. Experimental results showed that GraphShop outperformed the existing methods

Type	Distance to object location	Popularity	Rating	Average price
Fast food	50m	2.77	7	22
Fried dumpling	50m	2.48	7	nan
Fast food	50m	2.83	7	nan
Fast food	50m	1.61	7	21
Fast food	50m	1.95	7	nan
Fast food	100m	2.56	7	nan
Afternoon tea	100m	4.99	9	54
Afternoon tea	150m	5.36	8	31
Suzhou cuision	150m	5.74	9	235
Snack	200m	5.24	8	40
BBQ	200m	3.14	7	nan
Korean food	200m	5.92	8	35
Fast food	200m	4.19	7	20
Afternoon tea	200m	5.44	8	34
Fast food	250m	6.85	8	51
Dumpling	250m	5.66	8	36
Yunnan cuision	300m	7.37	9	77
Guilin noodles	300m	3.00	7	25

Table 6: The nearby shops of the available Malatang shop.

with a large margin.

We hope GraphShop and our open dataset could encourage the future study of this critical problem and help investors get insights and make our city a more habitable and convenient place. More useful data will be collected and analyzed soon, such as each shop's profit, the nearby institutes, and people's flow.

References

[1] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271, 2018.

[2] Yunsheng Bai, Hao Ding, Song Bian, Ting Chen, Yizhou Sun, and Wei Wang. Simgnn: A neural network approach to fast graph similarity computation. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pages 384–392, 2019.

[3] Wei-Ta Chu and Ya-Lun Tsai. A hybrid recommendation system considering visual information for predicting favorite restaurants. World Wide Web, 20(6):1313–1331, 2017.

[4] Lisette De Vries, Sonja Gensler, and Peter SH Leeflang. Popularity of brand posts on brand fan pages: An investigation of the effects of social media marketing. Journal of interactive marketing, 26(2):83–91, 2012.

[5] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In WWW, pages 417–426. ACM, 2019.

[6] Yanjie Fu, Hui Xiong, Yong Ge, Zijun Yao, Yu Zheng, and Zhi-Hua Zhou. Exploiting geographic dependencies for real estate appraisal: a mutual perspective of ranking and clustering. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1047–1056, 2014.

[7] Anant Gupta and Kuldeep Singh. Location based personalized restaurant recommendation system for mobile environments. In 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pages 507–511. IEEE, 2013.

[8] UN Habitat. State of the world's cities 2012/2013: Prosperity of cities. Routledge, 2013.

[9] Guy Hadash, Oren Sar Shalom, and Rita Osadchy. Rank and rate: multi-task learning for recommender systems. In RecSys, pages 451–454, 2018.

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Advances in neural information processing systems, pages 1024–1034, 2017.

[11] Xiangnan He, Ming Gao, Min-Yen Kan, Yiqun Liu, and Kazunari Sugiyama. Predicting the popularity of web 2.0 items based on user comments. In SIGIR, pages 233–242, 2014.

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In WWW, pages 173–182, 2017.

[13] Dmytro Karamshuk, Anastasios Noulas, Salvatore Scellato, Vincenzo Nicosia, and Cecilia Mascolo. Geo-spotting: mining online location-based services for optimal retail store placement. In KDD, pages 793–801. ACM, 2013.

[14] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. Computer, 42(8):30–37, 2009.

[15] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926, 2017.

[16] Xiaoxin Mao, Xi Zhao, Jun Lin, and Enrique Herrera-Viedma. Utilizing multi-source data in popularity prediction for shop-type recommendation. Knowledge-Based Systems, 165:253–267, 2019.

[17] Victor TC Middleton, Alan Fyall, Michael Morgan, and Ashok Ranchhod. Marketing in travel and tourism. Routledge, 2009.

[18] Wanli Min and Laura Wynter. Real-time road traffic prediction with spatio-temporal correlations. Transportation Research Part C: Emerging Technologies, 19(4):606–616, 2011.

[19] Beng S Ong. Online shopper reviews: Ramifications for promotion and website utility. Journal of Promotion Management, 17(3):327–344, 2011.

[20] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In CVPR, pages 652–660, 2017.

[21] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In The adaptive web, pages 291–324. Springer, 2007.

[22] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. IEEE transactions on acoustics, speech, and signal processing, 37(3):328–339, 1989.

[23] Lu Wang, Wei Zhang, Xiaofeng He, and Hongyuan Zha. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In KDD, pages 2447–2456. ACM, 2018.

[24] Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu. Exploring hierarchical structures for recommender systems. TKDE, 30(6):1022–1035, 2018.

[25] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In AAAI, volume 33, pages 346–353, 2019.

[26] Zechen Wu and Eisuke Ito. Correlation analysis between user's emotional comments and popularity measures. In 2014 IIAI 3rd International Conference on Advanced Applied Informatics, pages 280–283. IEEE, 2014.

[27] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In AAAI, 2018.

[28] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In KDD, pages 974–983. ACM, 2018.

[29] Zhiwen Yu, Miao Tian, Zhu Wang, Bin Guo, and Tao Mei. Shop-type recommendation leveraging the data from social media and location-based services. ACM Transactions on Knowledge Discovery from Data (TKDD), 11(1):1, 2016.

[30] Zhiwen Yu, Huang Xu, Zhe Yang, and Bin Guo. Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. IEEE Transactions on Human-Machine Systems, 46(1):151–158, 2015.

[31] Zhiyong Yu, Daqing Zhang, and Dingqi Yang. Where is the largest market: Ranking areas by popularity from location based social networks. In UIC, pages 157–162. IEEE, 2013.

[32] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR), 52(1):1–38, 2019.