Competitive Analysis for Points of Interest

Shuangli Li[1,2], Jingbo Zhou[2,3*], Tong Xu[1], Hao Liu[2,3], Xinjiang Lu[2,3], Hui Xiong[4*]
[1]School of Computer Science, University of Science and Technology of China, [2]Business Intelligence Lab, Baidu Research
[3]National Engineering Laboratory of Deep Learning Technology and Application, China, [4]Rutgers University
lsl1997@mail.ustc.edu.cn,{zhoujingbo,liuhao30,luxinjiang}@baidu.com,tongxu@ustc.edu.cn,hxiong@rutgers.edu

ABSTRACT

The competitive relationship of Points of Interest (POIs) refers to the degree of competition between two POIs for business opportunities from third parties in an urban area. Existing studies for competitive analysis usually focus on mining competitive relationships of entities, such as companies or products, from textual data. However, there are few studies which have a focus on competitive analysis for POIs. Indeed, the growing availability of user behavior data about POIs, such as POI reviews and human mobility data, enables a new paradigm for understanding the competitive relationships among POIs. To this end, in this paper, we study how to predict the POI competitive relationship. Along this line, a very first challenge is how to integrate heterogeneous user behavior data with the spatial features of POIs. As a solution, we first build a heterogeneous POI information network (HPIN) from POI reviews and map search data. Then, we develop a graph neural network-based deep learning framework, named DeepR, for POI competitive relationship prediction based on HPIN. Specifically, DeepR contains two components: a spatial adaptive graph neural network (SA-GNN) and a POI pairwise knowledge extraction learning (PKE) model. The SA-GNN is a novel GNN architecture with incorporating POI's spatial information and location distribution by a specially designed spatial oriented aggregation layer and spatial-dependency attentive propagation mechanism. In addition, PKE is devised to distill the POI pairwise knowledge in HPIN being useful for relationship prediction into condensate vectors with relational graph convolution and cross attention. Finally, extensive experiments on two real-world datasets demonstrate the effectiveness of our method.

KEYWORDS

Point of Interest; Graph Neural Networks; Competitive Analysis; Heterogeneous Information Network

ACM Reference Format:
Shuangli Li[1,2], Jingbo Zhou[2,3*], Tong Xu[1], Hao Liu[2,3], Xinjiang Lu[2,3], Hui Xiong[4*]. 2020. Competitive Analysis for Points of Interest. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394486.3403179

*Jingbo Zhou and Hui Xiong are corresponding authors. This work was done when Shuangli Li was an intern at the Baidu Research.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403179

1 INTRODUCTION

To sustain the prosperity of business, it is critical to providing an effective understanding of their competitive environment. For example, identifying competitors of a company can help to make a reasonable price level and appropriate management strategy for business. The study of competitive relationship prediction has been a hot research topic for a long time [4, 7]. Most of the existing works aim at solving the competition prediction between social events, companies or products [17, 25, 28, 29]. With the wide availability of Point-of-Interest (POI) data, we propose to investigate a new unique perspective for understanding the competitive environment by measuring the competition between POIs.

POI competitive relationship prediction aims to identify the degree of competition between two POIs to secure business from a third party in an urban area. POIs, like bars, retail stores, restaurants, and hotels, usually have to strive for limited users to survive. Therefore, being aware of the competitive relationship among POIs is fundamental to the shopkeeper of the POI to keep the business alive and thriving. Our study also has a lot of potential applications for location-based services, like POI recommendation, local marketing and location-based advertising. For example, the inferred competitive relationships can be used as features for advertising prediction models.

Previous studies of competitive relationship prediction cannot be applied to POIs. Most of the existing works focus on identifying comparative entities from sentences in text data like reviews, social networks and web pages [12, 17, 25]. However, such comparative evidence for POIs is often absent in text data. For example, it is quite rare to find the comparative sentence contains specific names of POIs (e.g., a KFC store) in POI reviews. Meanwhile, there is a large amount of user behavior data about POIs which can help to enable a new paradigm for understanding the relationships among POIs. If two POIs with similar functionalities have common users, they tend to be competitive. Besides, the user-generated reviews of a POI also contain many text descriptions of the POI. How to integrate such heterogeneous but useful information with the spatial features of POIs for competitive relationship prediction remains a unique research challenge.

In this paper, we propose a graph neural network-based (GNN-based) Deep learning framework for competitive Relationship prediction, named DeepR for short. The DeepR framework is running over a carefully designed heterogeneous POI information network (HPIN) which integrates the POI spatial feature, user behavior data and POI review data. To the best of our knowledge, we are the first to study the POI competitive relationship prediction problem.

The construction of HPIN enables us to investigate the POI competitive relationship prediction problem with heterogeneous data. In HPIN, we consider three context factors related to POI competitive relationship, which are spatial context, user behavior context, and aspect context. The spatial context represents the surrounding environment and relative positions of POIs in spatial space. The user behavior context captures relations among POIs hidden in the map search query data which records users' actions on POIs on an online map service platform, like Google Maps and Baidu Maps. Here we build a POI co-query graph where the node is a POI and an edge weight records how many users search the two POIs in a short time interval. The aspect context reflects the major services of a POI extracted from reviews. All these factors are integrated into a heterogeneous POI information network (HPIN) which is illustrated in Figure 2. A detailed explanation of HPIN will be introduced later.

The DeepR framework adopts the Graph Neural Network (GNN) as the basic model since GNN has achieved state-of-the-art performance for many relational learning problems. Particularly, due to the unique features of POIs, DeepR contains a novel spatial adaptive graph neural network (named SA-GNN) and a POI pairwise knowledge extraction learning (named PKE) model.

The SA-GNN is a novel GNN architecture tailored for POI competitive relationship prediction, which can handle the POI's spatial information and location distribution. Compared with other common graphs, an important feature of the POI graph (in HPIN) is that the relative spatial positions are important for competitive relationship because: 1) the nearby POIs are more likely to be competitive; 2) the spatial distribution of POIs also affects the result. For example, given a POI p_i, its competitive environment is more intense if all the competitors are evenly distributed around p_i, compared with the case that all the competitors are on one side of p_i. However, there are two drawbacks of traditional message-passing based GNN: 1) losing the spatial information of POIs and their neighbors; 2) lacking the ability to capture distant-range spatial location dependencies. These limitations degrade the performance of GNN for our problem. The novelty of SA-GNN relies on specially designed components upon GNN to handle such limitations. To address the first limitation, we propose a spatial oriented aggregation layer for SA-GNN; and for the second limitation, we devise a spatial-dependency attentive propagation mechanism for SA-GNN.

Another important part of DeepR is the PKE model which can distill the POI pairwise knowledge in HPIN for relationship prediction into condensate representation vectors. After extracting aspects from reviews, we build a relation-aware aspect graph convolutional network (RAConv) in HPIN to learn aspect embedding and brand embedding of POIs. Then a cross attention mechanism is designed to generate the aspect enhanced representation of POI pairs. The intuition behind the cross attention is to build the tailored representation of POI pairs with attention to the comparative aspects between a pair of POIs.

Finally, we unify all the components into the DeepR for POI competitive relationship prediction. To summarize, the main contributions of our work are as follows:

- We first study the POI competitive relationship prediction problem with heterogeneous information including human behavior data and POI review data. The POI competitive relationship prediction has great potential for many applications.
- We propose a novel DeepR framework over a carefully designed heterogeneous POI information network (HPIN). Several techniques, such as SA-GNN, spatial oriented aggregation layer, spatial-dependency attentive propagation and PKE model for knowledge extraction, are invented to handle the unique properties of POIs for competitive relationship prediction.
- Extensive experiments are conducted on two real-world datasets, demonstrating the effectiveness of our proposed DeepR model.

2 RELATED WORK

The research topic of this paper is closely related to the competitive relationship mining, link prediction and graph neural networks. In this section, we will briefly discuss them respectively.

Competitive Relationship Mining. Most of the existing studies aim at detecting the competitive relationship between companies or products on text data. The early research [4, 7] applies predefined linguistic patterns (i.e., A vs B) for mining competitors. A graphical model in [25] is designed to extract product comparative relations from reviews. The authors in [12] propose a network-based approach based on company citations (co-occurrence) graph in online news articles. CMiner [17], which compares specific features on user reviews, is useful for explaining the reasons for competition. TFGM [26] learns from patent and twitter data to classify the relationships between entities by a topic model. All the above methods cannot handle the unique features of POIs which have spatial context and user behavior context factors. In a word, there are few works on mining competitive relationship for Points of Interest.

Link Prediction. There are many methods for solving the link prediction problem in different fields, including the statistical methods [8, 21] and graph embedding methods [2, 11, 15]. Some works also study the link prediction on heterogeneous information networks by the auto-encoder model [20] and the meta path-based methods [3]. Notice that the above methods mostly treat the relationship prediction as completing the missing links, with a strong assumption that all the target links to predict are exactly in the graph. However, our problem is not a link completion problem since what we predict is the POI competitive relationship which does not exist in HPIN. Therefore, the above methods lose their effectiveness. Furthermore, they cannot consider additional information such as spatial information and user behavior information.

Graph Neural Networks. Recently, many efforts have been devoted to study Graph Neural Networks (GNNs)[6, 14, 18, 19, 22, 31]. HAN[22] uses the hierarchical attention to aggregate feature information through the predefined meta-paths in heterogeneous graphs. There are also some works designed for link prediction [16, 30]. SEAL[30] predicts the general relationship on the graph based on graph neural network and Weisfeiler-Lehman neural machine. The above GNN-based models can only capture the topological structure features of the graph, but the necessary spatial information and aspect information of POIs can not be handled by the existing models. In recent years the GNN and graph embedding methods have also been applied to solve spatio-temporal problems in specific applications [9, 10, 24, 32]. There are also some works about so-called spatial-temporal graph neural network. However, these spatial graph neural networks mainly refer to the message passing process over the original "node" space (instead of graph spectral space) which convolves the central node's representation with its neighbors' representations to derive the updated representation for the central node [23].

3 OVERVIEW

In this section, we first introduce the preliminaries, then we present a framework overview to show the work process of the competitive relationship prediction for POIs.

3.1 Preliminaries

Here we first provide some basic concepts used in our paper, following by a formal definition of our problem.

Definition 3.1. **Heterogeneous POI Information Network.** The Heterogeneous POI Information Network (HPIN) is defined as $G = (\mathcal{P} \cup \mathcal{B} \cup \mathcal{A} \cup \mathcal{M}, \mathcal{E}_{pp} \cup \mathcal{E}_{pb} \cup \mathcal{E}_{bb} \cup \mathcal{E}_{ba} \cup \mathcal{E}_{aa})$, where $\mathcal{P}, \mathcal{B}, \mathcal{A}$, and $\mathcal{M}$ are the sets of nodes in HPIN. In specific, $\mathcal{P} = \{p_1, ..., p_{n_p}\}$ denotes the set of POIs, $\mathcal{B} = \{b_1, ..., b_{n_b}\}$ denotes the set of brands, $\mathcal{A} = \{a_1, ..., a_{n_a}\}$ denotes the set of aspects extracted from reviews, and $\mathcal{M} = \{\mathcal{M}_1, ..., \mathcal{M}_{n_M}\}$ denotes the set of heat maps. n_p, n_b, n_a, n_M are the number of POI, brand, aspect, spatial heat map in HPIN. Each POI node p_i is associated with a heat map to represent the surrounding distribution feature of POI. $\mathcal{E}_{pp}$, $\mathcal{E}_{pb}$, $\mathcal{E}_{bb}$, $\mathcal{E}_{ba}$ and $\mathcal{E}_{aa}$ are five different sets of relational edges (POI-POI, POI-brand, brand-brand, brand-aspect, and aspect-aspect) among the three types of nodes. How to construct HPIN (including $\mathcal{M}$) is introduced in Section 4.

Definition 3.2. **Meta-path for Brand.** We define meta-path Φ as a path with the form $b_i \xrightarrow{R_{pb}^{-1}} p_k \xrightarrow{R_{pp}} p_l \xrightarrow{R_{pb}} b_j$, which describes a POI-based brand correlation between brand b_i and brand b_j via a path between POI p_k and POI p_l.

Taking Figure 2 for example, there are four POIs in the HPIN, each POI belongs to a brand, denoted as an edge in $\mathcal{E}_{pb}$, meaning that a POI is one branch of its brand (i.e., p_2 is one branch of KFC). In addition, the attributes of POI include its category (i.e., the category of p_2 is the fast-food restaurant) and coordinate. The relation between POIs is based on the user map search query. For example, p_1 is linked with p_2 in HPIN because they are always queried together by users on the online map. The relation between brands is defined based on meta-path Φ. In Figure 2(c), the edge b_2b_3 is constructed by the path $b_2 \rightarrow p_2 \rightarrow p_3 \rightarrow b_3$. Each brand is also associated with several aspects. b_3 (Mcdonald's) is closely related to a_2 (French Fries) and a_3 (hamburger). The two aspects a_2 and a_3 often appear together in the user reviews so they are connected with each other.

Definition 3.3. **Competitive Relationship Prediction.** The objective of our problem is to associate each POI pair (p_i, p_j) with a label $y \in \{0, 1\}$ where $y = 1$ indicates p_i and p_j have a competitive relationship. Formally, given a set of POIs $\mathcal{P}$ and an HPIN G, the goal is to learn a predictive function $f : (\mathcal{P} \times \mathcal{P} \mid G) \rightarrow Y$ to predict the competitive relationship between POIs.

3.2 Framework Overview

Figure 1 shows an overview of the POI competitive relationship analysis process. First, we construct a HPIN from map query data and user review data with POI attributes. Then we input the aspect graph and POI graph extracted from the HPIN to the proposed DeepR model, and the outputs indicate whether the given POI pairs have competitive relationships.

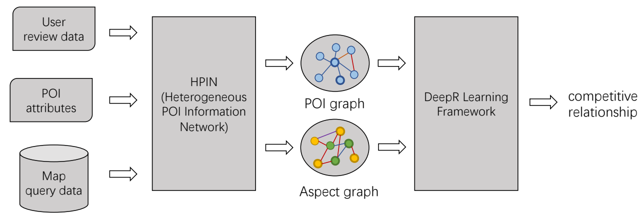


Figure 1: An overview of the POI competitive relationship analysis process.

Figure 3 presents the whole framework of DeepR, which consists of two main components: *Spatial Adaptive Graph Neural Network (SA-GNN) Learning* and *Pairwise POI Knowledge Extraction* (PKE). First, SA-GNN aggregates the neighbors on the POI graph with integrating the spatial information via the spatial oriented aggregation layer, and then attentively propagates the distant-range dependencies with spatial locations. Next, SA-GNN is applied to two sub-graphs (Diffusion and Affinity Graph) of the POI graph to capture the hidden patterns indicated by category information on the POI graph. Second, PKE adopts the *relation-aware aspect graph convolution* (RAConv) to encode the brand and aspect on the aspect graph. Then PKE employs the cross attention mechanism to distill the pairwise POI knowledge from aspects, which is combined with the POI pair representation next. Finally, the pairwise prediction part outputs the result of the competition. In the following two sections, we first introduce the construction of HPIN, then we present our DeepR framework.

4 HPIN CONSTRUCTION

In this section, we describe how to construct the HPIN with an example of HPIN shown in Figure 2. We first introduce the method of constructing spatial heat maps linked with POIs. Then we discuss the process of building different relations in HPIN. The relation of POI-brand is given by the attributes of POI, as illustrated in Table 2 in appendix, showing which brand every POI belongs to. We mainly introduce the POI relation (POI-POI), and the relations of aspect and brand (brand-aspect, aspect-aspect and brand-brand).

4.1 Spatial Heat Map Construction

As illustrated in Figure 2(a), $\mathcal{M} = \{\mathcal{M}_1, ..., \mathcal{M}_{n_M}\}$ is a set of grid spatial heat maps that are linked with POIs to represent the surrounding environment features. Each element $\mathcal{M}_i$ of $\mathcal{M}$ corresponds to a spatial heat matrix $\boldsymbol{M}_i \in \mathbb{R}^{C \times L \times L}$, where C is the number of category-based channels and L is the side length of the matrix. We split a city into the same size grids (i.e., 500m×500m). Each grid S_k has C channels respecting to the number of categories of POIs, which can be denoted as $\{\mathbf{v}_k^1, ..., \mathbf{v}_k^C\}$. The value of $\mathbf{v}_k^c$ is calculated with max-pooling method:

$$\mathbf{v}_k^c = \mathbf{max}_{\forall p_t \in S_k} \Big\{ f_{hot}(p_t) \mid \text{tag}(p_t) = c, 1 \leq c \leq C \Big\}, \quad (1)$$

where $\text{tag}(p_t) = c$ limits p_t has a category of c, and f_{hot} returns the hot value of p_t, which is total times of p_t being searched by users recorded on map search query data in a time interval. We treat one central grid of each POI with its neighbors as $L \times L$ grids to constitute the spatial heat map.

4.2 POI Relation Construction

POI co-query edge in $\mathcal{E}_{pp}$ is a type of behavior-driven relationship. Here we use map search query data which records users' actions on POIs on an online map service platform to construct the behavior-driven relationship. It is also possible to use other similar user behavior data. Intuitively the more frequently a pair of POIs (p_i, p_j) are queried by the same users in a short time interval Δt, the more likely they are related to each other. Each POI-POI edge $(p_i, p_j, w^q_{ij}) \in \mathcal{E}_{pp}$ means that p_i and p_j are queried w^q_{ij} times together by all users in a period of time. We use $\mathbf{A}_q$ to denote the adjacency matrix of the POI co-query, and $(\mathbf{A}_q)_{ij} = (\mathbf{A}_q)_{ji} = w^q_{ij}$. To reduce noise disturbance and reduce the size of the co-query graph, we set a threshold θ_m to filter edges that satisfy $w^q_{ij} < \theta_m$. We also name the graph formed by POIs and edges $\mathcal{E}_{pp}$ as POI co-query graph.

The POI co-query graph can also be divided into two sub-graphs: a diffusion graph and an affinity graph. If a pair of POIs with the same category are searched many times, they tend to be competitive since users may make a choice between them; if ones with different categories are searched many times, they tend to be complementary (instead of competitive) since users may plan to visit both of them. Due to the difference of behavior semantics, we build a diffusion graph on the same category of POIs, and an affinity graph on the different categories of POIs. As introduced later, SA-GNN will be applied to the diffusion graph and the affinity graph separately.

4.3 Aspect and Brand Relation Construction

Aspect set $\mathcal{A}$ is extracted from reviews data. We first gather all user reviews of a brand to format a document. The reasons to gather reviews to brands instead of POIs are: 1) the reviews of a certain POI are usually rare; 2) most users give reviews about the brand (like "KFC") instead of a certain store. If a POI does not have a brand, we use its name as a brand. We treat brands as documents and aspects as the key words of documents. Then we calculate words' term frequency-inverse document frequency (TF-IDF), and we select the top k words as aspects in aspect set, denoted as $a_j \in \mathcal{A}$. In this way, brand-aspect relation can also be built, $(b_i, a_j, w^t_{ij}) \in \mathcal{E}_{ba}$ means that a_j is one of the top k aspects of b_i and edge weight w^t_{ij} is the TF-IDF value. We use $(\mathbf{A}_t)_{ij} = w^t_{ij}$ to denote the adjacency matrix of $\mathcal{E}_{ba}$. We also employ point-wise mutual information (PMI) to establish linkages between aspects [27]. Thus, we have $PMI(i, j) = w^a_{ij}$ to measure the relevance of aspects (a_i, a_j), denoted as $(a_i, a_j, w^a_{ij}) \in \mathcal{E}_{aa}$, and $(\mathbf{A}_a)_{ij} = w^a_{ij}$. We filter out aspect pairs whose PMI is lower than a threshold θ_{PMI}.

In addition, there is an edge between brands too. We count the number of path Φ from b_i to b_j and normalize the result:

$$s(b_i, b_j) = s(b_j, b_i) = \frac{\left|\{\boldsymbol{p}_{b_i \rightsquigarrow b_j} : \boldsymbol{p}_{b_i \rightsquigarrow b_j} \models \Phi\}\right|}{\sqrt{\left|\mathcal{N}^{(pb)}_i\right|} \cdot \sqrt{\left|\mathcal{N}^{(pb)}_j\right|}}, \tag{2}$$

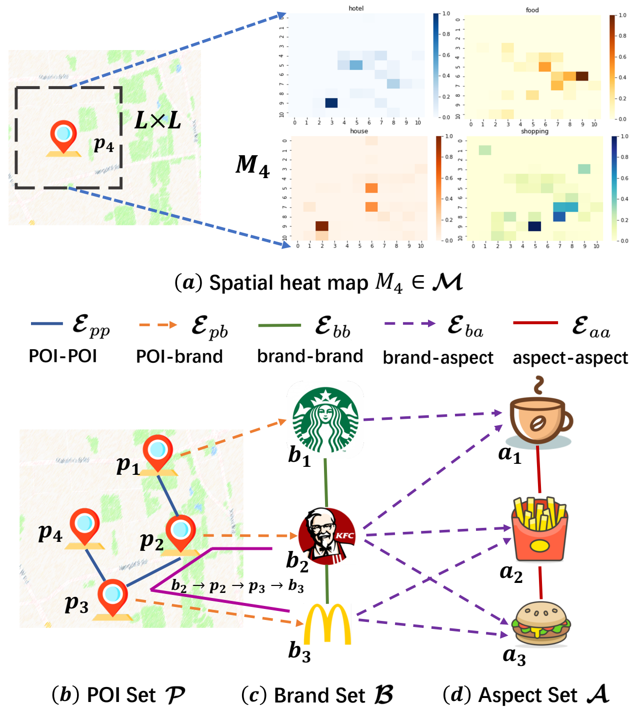


Figure 2: An illustrative example of HPIN. (a) Spatial heat map of several category-based channels (i.e., hotel, restaurant, shopping center, house). (b) POI set. (c) Brand (i.e., KFC, Mcdonald's, Starbucks) set. (d) Aspect (i.e., coffee, French Fries, hamburger) set.

where $\models$ means path $\boldsymbol{p}_{b_i \rightsquigarrow b_j}$ follows the rule defined by meta-path Φ, and $\left|\mathcal{N}^{(pb)}_i\right|$ denotes the number of POIs belonging to brand b_i. Formally, edges of brands are $(b_i, b_j, w^b_{ij} = s(b_i, b_j)) \in \mathcal{E}_{bb}$, and adjacency matrix as $(\mathbf{A}_b)_{ij} = w^b_{ij}$.

5 DEEPR FRAMEWORK

In this section, we present our DeepR framework whose architecture is illustrated in Figure 3. We first introduce two important components of DeepR, which are spatial adaptive graph neural network (SA-GNN) learning component and pairwise POI knowledge extraction model (PKE). Then we apply a pairwise output layer to predict the competitive probability given a pair of POIs.

5.1 Spatial Adaptive Graph Neural Network

Though graph neural networks have been widely studied in many applications and show great advantages in the general graph learning problem, these message-passing neural networks (MPNNs), such as GCN [6] and HAN [22], still have fundamental weaknesses in several respects [14], especially in our POI graph learning problem. Firstly, the aggregation of MPNNs treats all neighbors of POIs equally in direction and loses the spatial information of POIs and their neighbors. Secondly, spatial distance dependencies are neglected, causing the lack of the ability to capture distant-range spatial location dependencies.

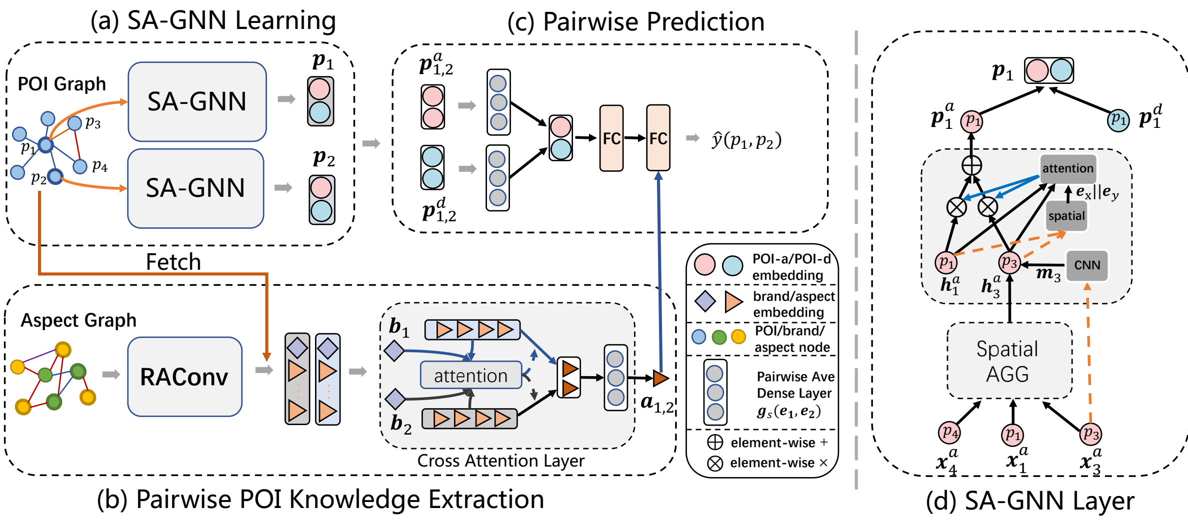


Figure 3: Illustration of the proposed DeepR framework. We input the POI graph and aspect graph extracted from the heterogeneous POI information network (HPIN). Figures (a) (b) (c) show different components, and Figure(d) presents the inner structure of the SA-GNN component.

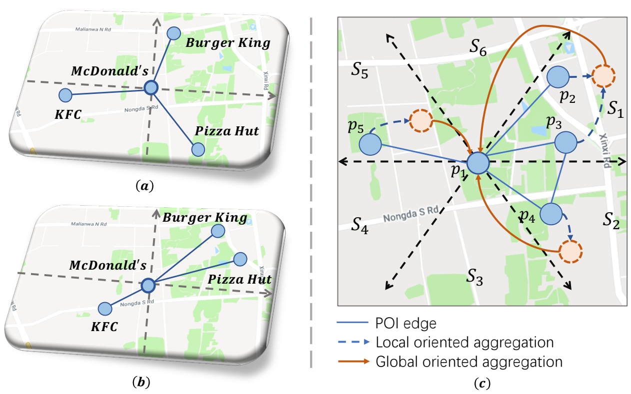


Figure 4: (a,b) Two types of spatial distribution example. (c) Explanation of the spatial oriented aggregation process of SA-GNN.

For example, as shown in Figure 4, McDonald's has three neighbors on the POI graph. In Figure 4(a), three neighbor POIs relatively evenly distributed around McDonald's. In this case, three POIs probably have the same effect on McDonald's from the point of the spatial distribution. In Figure 4(b), KFC is located on one side of McDonald's, while Burger King and Pizza Hut are on another side. What's more, compared to Burger King and Pizza Hut, KFC is much closer to McDonald's in distance. Thus McDonald's should pay more attention to KFC than that to Burger King and Pizza Hut, because people who live on the side of KFC are more likely to make a choice between KFC and McDonald's. But MPNNs can not tell the difference between Figure 4(a) and Figure 4(b).

To overcome the above two limitations of traditional GNNs based on message-passing, as shown in Figure 3(d), we design the SA-GNN to process both the user behavior context and the spatial context of POI in HPIN with considering spatial location relation on the co-query graph and spatial distribution of POIs. SA-GNN is composed of spatial oriented aggregation layer and spatial-dependency attentive propagation layer.

5.1.1 Spatial Oriented Aggregation Layer. To overcome the first limitation and aggregate POI neighbors on the co-query graph spatially, we adopt a graph convolutional layer to learn POI representations by considering spatial location relation, as illustrated in Figure 4(c). We first evenly divide the neighbors of each POI node p_i into several sectors (e.g., six sectors) $S_1, ..., S_n$ according to their location coordinates. Each POI in the sector S_k belongs to the same spatial neighbor set of p_i, denoted as $\mathcal{N}_k(p_i)$. POIs in the same sector S_k are associated with the spatial relationship sr_k. Notice that p_i itself does not belong to any sector $S_1 \sim S_n$, so we define the central node p_i in the sector S_0 with neighbor set $\mathcal{N}_0(p_i) = \{p_i\}$. Then we use the same graph convolutional rule based on symmetric normalized Laplacian as GCN [6] to aggregate POIs of the relationship sr_k locally in the sector S_k:

$$s_i^k = \sum_{p_j \in \mathcal{N}_k(p_i)} (\deg(p_i)\deg(p_j))^{-\frac{1}{2}} \boldsymbol{x}_j, \tag{3}$$

where $\deg(p_i)$ is the degree of node p_i in co-query graph. Then we aggregate the representations of different sectors for each POI:

$$\boldsymbol{q}_i = \sigma(\boldsymbol{W}_q \cdot \mathop{||}_{k=0}^{n} s_i^k), \tag{4}$$

where σ is the non-linear activation function, $\boldsymbol{W}_q$ is the transform matrix and we use concatenation operator || as the global oriented aggregation function. In this spatial oriented aggregation way, SA-GNN can distinguish the POI neighbors from different sectors and integrate spatial information.

5.1.2 Spatial-dependency Attentive Propagation Layer. To overcome the second limitation and capture distant-range dependencies, we propose a spatial attentive propagation layer to handle the spatial information with two techniques. First, we design a heat map convolution layer to model the surrounding environment of POIs to capture the spatial distribution feature. For a target POI p_i, we input its spatial heat map matrix in HPIN, denoted as $\boldsymbol{M}_i$, into CNN operation and learn the distant-range dependencies feature $\boldsymbol{m}_i$:

$$\boldsymbol{m}_i = f_{CNN}\Big(\boldsymbol{M}_i; \boldsymbol{w}_h\Big), \tag{5}$$

Then we concatenate $\boldsymbol{m}_i$ and $\boldsymbol{q}_i$ to learn POI representation feature:

$$\boldsymbol{h}_i = \mathrm{CONV}(p_i) = \sigma\big(\boldsymbol{q}_i \oplus \boldsymbol{m}_i\big), \tag{6}$$

Second, we invent a location-aware attentive propagation layer to process the relative spatial positions between POIs. Although the spatial oriented aggregation layer can capture POI spatial information of different sectors, the spatial distance factor between POIs is overlooked. As illustrated in Figure 4, the nearer a neighbor POI is to another POI, the more possible they are competitive, which means this POI deserves higher attention in the model. Besides, the relative spatial location in two-dimensional space is also influential. To overcome this problem, we propose a location-aware attention mechanism to propagate POI representation feature:

$$\boldsymbol{p}_i = \sum_{j \in \mathcal{N}_i} attn_s(p_i, p_j, \boldsymbol{r}_s)\boldsymbol{W}_p \cdot \mathrm{CONV}(p_j), \tag{7}$$

where $attn_s(p_i, p_j, \boldsymbol{r}_s)$ is the location-aware attentive weight of each neighbor p_j for p_i with their relative spatial location feature $\boldsymbol{r}_s$. $\boldsymbol{W}_p$ is a transform matrix.

As shown in Figure 5, We define two one-hot vectors $\boldsymbol{a}_x(p_i, p_j)$ and $\boldsymbol{a}_y(p_i, p_j)$, which represent the relative position in longitude and latitude dimensions. For each dimension, we take the p_i as the origin of coordinates and divide the distance between p_i and p_j in two dimensions into buckets (in our experiment, we set the bucket as 100 meters, and the maximum distance in each dimension is 10km). Then we concatenate two embeddings $\boldsymbol{e}_x(i,j) = \boldsymbol{W}_x \boldsymbol{a}_x(p_i, p_j)$ and $\boldsymbol{e}_y(i,j) = \boldsymbol{W}_y \boldsymbol{a}_y(p_i, p_j)$, and we apply a dense layer transformation:

$$\boldsymbol{r}_s = \boldsymbol{W}_s \cdot \big(\boldsymbol{e}_x(i,j) \oplus \boldsymbol{e}_y(i,j)\big), \tag{8}$$

Next, We implement location-aware attention $attn_s$, which is formulated as shown:

$$attn_s(p_i, p_j, \boldsymbol{r}_s) = \sigma\big(\boldsymbol{a}^T \cdot (\boldsymbol{W}_t\, \boldsymbol{h}_i \oplus \boldsymbol{W}_t\, \boldsymbol{h}_j \oplus \boldsymbol{r}_s)\big), \tag{9}$$

Finally, as suggested in GAT [18], multi-head attention is used to enhance learning ability during propagation. We concatenate K embeddings learned by K independent location-aware attention mechanisms to represent final aggregation embedding:

$$\boldsymbol{p}_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} attn_s^k(p_i, p_j, \boldsymbol{r}_s)\boldsymbol{W}_p^k \cdot \mathrm{CONV}(p_j)\right).$$

5.1.3 Learning on Diffusion and Affinity Graphs. We further apply SA-GNN on the two sub-graphs deriving from co-query graph. As we have stated, on co-query graph, there are two sub-graphs which are diffusion graph and affinity graph. The diffusion graph is based on POIs with the same category (which tend to be competitive since users may make a choice among them), and the affinity graph is based on POIs with different categories (which are complementary instead of competitive since users may plan to visit all of them). Due

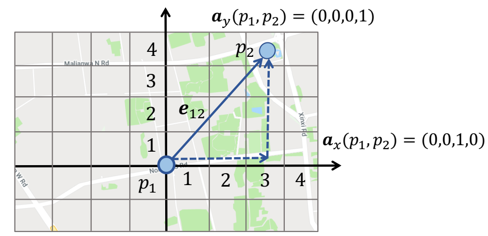


Figure 5: Relative spatial location feature of p_1 and p_2

to the different behavior semantics behind this view, we apply SA-GNN operation to these two different graphs separately, denoted as $\boldsymbol{p}_i^d$ and $\boldsymbol{p}_i^a$, and concatenate two types of representations to output final POI p_i representation feature:

$$\boldsymbol{p}_i = \boldsymbol{p}_i^d \oplus \boldsymbol{p}_i^a, \tag{10}$$

5.2 Pairwise POI Knowledge Extraction

DeepR also utilizes the aspect information and brand information of POI in HPIN to help competitive relationship prediction. Formally, Given a pair of POIs into DeepR, (Note that one POI in HPIN, e.g., “KFC in West Court”, corresponds to one brand, e.g., “KFC”), we can fetch a pair of brand embeddings, denoted as $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$. Such embedding features can be used to enhance the POIs for relationship prediction. To meet it, as shown in Figure 3(b), we present the other component of DeepR: Pairwise POI Knowledge Extraction (PKE), which consists of relation-aware aspect convolution (RAConv) and cross attention layer. In this section, We discuss how to generate the embeddings of brands and aspects, and then distill the pairwise POI knowledge about the aspect for each POI pair from HPIN.

5.2.1 Relation-aware Aspect Convolution. Inspired from Text GCN [27] and R-GCN [16], we propose relation-aware aspect convolution (RAConv) to learn brand and aspect embeddings which can distinguish three relations of brand and aspect (brand-brand relation, aspect-aspect relation and brand-aspect relation). For each brand and aspect, we use relation-aware graph convolution operation to aggregate all related neighbor brands and aspects:

$$\begin{aligned} AGG(\boldsymbol{a}_i^{(l)}) &= \sum_{j \in \mathcal{N}_i^a} (\hat{\mathrm{A}}_a)_{ij} \boldsymbol{W}_a \boldsymbol{a}_j^{(l-1)} + \sum_{j \in \mathcal{N}_i^t} (\hat{\mathrm{A}}_t)_{ij} \boldsymbol{W}_t \boldsymbol{b}_j^{(l-1)}, \\ \boldsymbol{a}_i^{(l)} &= \sigma\Big(\boldsymbol{W}\boldsymbol{a}_i^{(l-1)} + AGG(\boldsymbol{a}_i^{(l)})\Big), \end{aligned} \tag{11}$$

$$\begin{aligned} AGG(\boldsymbol{b}_i^{(l)}) &= \sum_{j \in \mathcal{N}_i^b} (\hat{\mathrm{A}}_b)_{ij} \boldsymbol{W}_b \boldsymbol{b}_j^{(l-1)} + \sum_{j \in \mathcal{N}_i^t} (\hat{\mathrm{A}}_t)_{ij} \boldsymbol{W}_t \boldsymbol{a}_j^{(l-1)}, \\ \boldsymbol{b}_i^{(l)} &= \sigma\Big(\boldsymbol{W}\boldsymbol{b}_i^{(l-1)} + AGG(\boldsymbol{b}_i^{(l)})\Big), \end{aligned} \tag{12}$$

where the function AGG denotes that each aspect $\boldsymbol{a}_i$ or brand $\boldsymbol{b}_i$ aggregates the influences from its related brands and aspects; $\boldsymbol{a}_i$ represents aspect embedding and $\boldsymbol{b}_i$ represents brand embedding; $\hat{\mathrm{A}}_a = \widetilde{\mathrm{D}}_a^{-\frac{1}{2}} \widetilde{\mathrm{A}}_a \widetilde{\mathrm{D}}_a^{-\frac{1}{2}}$, $\widetilde{\mathrm{A}}_a = \mathrm{A}_a + \mathrm{I}_N$ ($\hat{\mathrm{A}}_b, \hat{\mathrm{A}}_t$ in the same way); $\mathcal{N}_i^a$, $\mathcal{N}_i^b$ and $\mathcal{N}_i^t$ denote the neighbor indices set of node i in three relations; Here σ is a non-linear activation; $\boldsymbol{W}$, $\boldsymbol{W}_a$, $\boldsymbol{W}_b$ and $\boldsymbol{W}_t$ are weight matrices of different relations.

5.2.2 *Cross Attention.* Given a pair of brand embeddings generated by the above RAConv, denoted as $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$, there are two lists of aspect embeddings, denoted as $\{\boldsymbol{a}_1^i, ..., \boldsymbol{a}_m^i\}$ and $\{\boldsymbol{a}_1^j, ..., \boldsymbol{a}_n^j\}$. Then we feed the embeddings into the cross attention layer to mutually generate attentive weights. We first calculate the similarity between brand b_i and each aspect a_l^j of brand b_j, denoted as $\boldsymbol{\pi}(b_i, a_l^j)$, aiming to pick up the important aspects and de-emphasize the noisy aspects of another brand b_j (and $\boldsymbol{\pi}(b_j, a_k^i)$ is the same):

$$\boldsymbol{\pi}(b_i, a_l^j) = \frac{\boldsymbol{b}_i \cdot \boldsymbol{a}_l^j}{\|\boldsymbol{b}_i\| \cdot \|\boldsymbol{a}_l^j\|}, l \in [1, n] \tag{13}$$

Then softmax function is used to obtain each other's normalized attentive weights of aspects, and the aspect enhanced feature vector $\boldsymbol{a}_i$ for POI p_i can be computed as the weighted sum of brand b_i's aspect a_k^i:

$$\boldsymbol{a}_i = \sum_{k=1}^{m} \beta_k \boldsymbol{a}_k^i, \tag{14}$$

$$\beta_k = \frac{\exp(\boldsymbol{\pi}(b_j, a_k^i))}{\sum_{t=1}^{m} \exp(\boldsymbol{\pi}(b_j, a_t^i))}, \tag{15}$$

Definition 5.1. **Pairwise Average Dense Operation.** We define a pairwise average dense operation $\boldsymbol{g}_s$ to overcome the asymmetry of concatenation two vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ directly:

$$\boldsymbol{g}_s(\boldsymbol{e}_1, \boldsymbol{e}_2) = \mathbf{average}\big(\boldsymbol{W}_g(\boldsymbol{e}_1 \oplus \boldsymbol{e}_2), \boldsymbol{W}_g(\boldsymbol{e}_2 \oplus \boldsymbol{e}_1)\big), \tag{16}$$

Given a pair of POIs, we input $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ into $\boldsymbol{g}_s$ to output final pairwise aspect representation $\boldsymbol{a}_{i,j}$:

$$\boldsymbol{a}_{i,j} = \boldsymbol{g}_s(\boldsymbol{a}_i, \boldsymbol{a}_j). \tag{17}$$

5.3 Prediction and Optimization

For a pair of POIs p_i, p_j, DeepR encodes p_i, p_j separately by SA-GNN layer and outputs their representation features, denoted as $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. After pairwise POI knowledge extraction, $\boldsymbol{a}_{i,j}$ represents the pairwise aspect context feature of p_i and p_j.

5.3.1 *Pairwise Interaction.* Pairwise interaction layer combines POI representation feature of p_i, p_j and outputs POI pair feature for competitive relationship prediction:

$$\boldsymbol{p}_{i,j} = \boldsymbol{W}_t \cdot \big(\boldsymbol{g}_s(\boldsymbol{p}_i^a, \boldsymbol{p}_j^a) \oplus \boldsymbol{g}_s(\boldsymbol{p}_i^d, \boldsymbol{p}_j^d)\big), \tag{18}$$

where $\boldsymbol{W}_t$ is the weight matrix, and $\boldsymbol{p}_i^d, \boldsymbol{p}_i^a$ are two POI representations of diffusion graph and affinity graph.

5.3.2 *Prediction.* Finally, the probability of a pair of POIs competing with each other is predicted by a fully connected layer with concatenating POI representation feature $\boldsymbol{p}_{i,j}$ and aspect context feature $\boldsymbol{a}_{i,j}$. Then we have:

$$\hat{y}_{i,j} = \mathbf{sigmoid}(\boldsymbol{W}_o \cdot (\boldsymbol{p}_{i,j} \oplus \boldsymbol{a}_{i,j})), \tag{19}$$

5.3.3 *Optimization.* We adopt the Cross-Entropy loss function to train DeepR over all labeled POI pairs between the ground-truth and the prediction:

$$\mathcal{L} = \sum_{(p_i,p_j) \in \mathcal{D}} \big(y_{i,j} log\hat{y}_{i,j} + (1 - y_{i,j})log(1 - \hat{y}_{i,j})\big), \tag{20}$$

where $\mathcal{D}$ is the ground-truth set and includes positive and negative labels of the competitive relationship. $y_{i,j}$ is the label of the pair (p_i, p_j) and $\hat{y}_{i,j}$ is the prediction result of DeepR.

6 EXPERIMENTS

In this section, we first introduce the experiment settings, and then demonstrate the effectiveness of DeepR on two city-level datasets.

6.1 Experiment Settings

6.1.1 *Dataset.* Experiments are conducted on two real-world POIs datasets in **Beijing** and **Chengdu**. To construct the HPIN, we use the map search query data and POI data in the corresponding cities from 1st August 2018 to 31st August 2018. The dataset is a portion of the whole data of **Beijing** and **Chengdu** randomly sampled from the data of Baidu Maps. We extract 3,231 aspects from reviews. And there are 113,997 links of aspect-aspect relation, 33,098 links of brand-brand relation and 207,917 links of brand-aspect relation.The construction of ground-truth is introduced in Appendix A.1.

6.1.2 *Baselines and Evaluation Metrics.* We use five kinds of baselines including the simple rule-based methods (**DIST** and **EW**), feature-based methods (**MLP** and **XGboost**[1]), graph embedding methods (**Deepwalk**[15] and **Node2vec**[2]), several state-of-the-art GNN models (**GCN**[6], **GAT** [18], **SEAL**[30], **Geom-GCN**[14]) and state-of-the-art GNN over heterogeneous information network(**HAN**[22]). The detail of baselines is introduced in Appendix A.2. We use Accuracy (Acc), Area Under Curve (AUC), Precision (Prec), Recall (Rec) and F1-score (F1) as the evaluation metrics.

6.2 Performance Evaluation

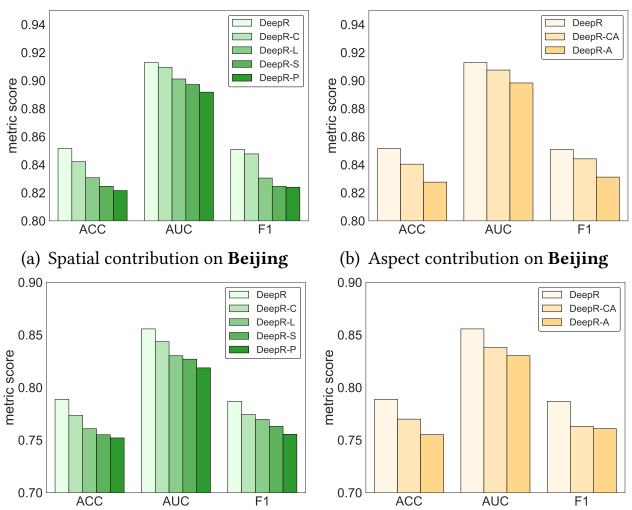
6.2.1 *Overall Comparison.* Table 1 shows the performance results of our proposed DeepR as compared to all the baselines on Beijing and Chengdu datasets. As we can see, DeepR significantly outperforms all the baselines in almost all metrics.

Specifically, we can observe that the prediction results of the rule-based methods (DIST, EW) are very poor, especially the accuracy is only around 60%, demonstrating that we cannot simply use distance or co-query weight to make predictions. Furthermore, feature-based models (MLP, XGBoost) are also not ideal compared to DeepR due to the lack of the incorporation of the co-query graph structure information. For graph embedding models (Deepwalk, Node2vec), they can learn the structure features of the POI co-query graph. As a result, they achieve better performances compared with the feature-based models. But the accuracy is still not high, because these graph embedding models fail to capture effective neighborhood information as graph neural networks.

For graph neural networks, GCN and GAT outperform relatively better than graph embedding baseline methods, which indicates the powerful ability of GNN to capture the graph information with POI features. The performance of Geom-GCN is not significantly better than GCN and GAT, the potential reason is that neighborhood graph structure and latent dependency in graph are not essential for competitive relationship prediction on our datasets. We notice that the result of SEAL is not as good as other GNN models, because the POI graph is much sparser than general graphs and SEAL has the limited effect to learn the subgraph pattern for link prediction. Since HAN can learn the heterogeneous information compared to

Table 1: Experimental results on Beijing and Chengdu dataset.

	Beijing					Chengdu				
	Acc	AUC	F1	Prec	Rec	Acc	AUC	F1	Prec	Rec
EW	0.5765	0.6225	/	/	/	0.5667	0.6133	/	/	/
DIST	0.6442	0.7131	/	/	/	0.6257	0.6963	/	/	/
MLP	0.7221	0.8102	0.7389	0.6968	0.7863	0.6883	0.7476	0.7117	0.6621	0.7694
XGboost	0.7814	0.8641	0.7915	0.7566	0.8298	0.7300	0.8090	0.7353	0.7211	0.7500
Deepwalk	0.7732	0.8511	0.7811	0.7549	0.8511	0.7397	0.8158	0.7485	0.7241	0.7745
Node2vec	0.7784	0.8527	0.7866	0.7586	0.8167	0.7411	0.8151	0.7518	0.7291	0.7759
GCN	0.8061	0.8790	0.8139	0.7826	**0.8477**	0.7534	0.8394	0.7569	0.7463	0.7677
GAT	0.8069	0.8828	0.8077	0.8046	0.8108	0.7581	0.8281	0.7542	0.7669	0.7418
Geom-GCN	0.8091	0.8835	0.8045	0.8071	0.802	0.7527	0.8309	0.7447	0.7697	0.7213
SEAL	0.8023	0.8813	0.8094	0.7814	0.8396	0.7489	0.8418	0.7505	0.7455	0.7557
HAN	0.8145	0.8893	0.8175	0.8046	0.8308	0.7633	0.8424	0.7656	0.7556	0.7758
DeepR	**0.8516**	**0.9129**	**0.8509**	**0.8546**	0.8472	**0.7876**	**0.8566**	**0.7884**	**0.7857**	**0.7911**



(a) Spatial contribution on **Beijing** (b) Aspect contribution on **Beijing**

(c) Spatial contribution on **Chengdu** (d) Aspect contribution on **Chengdu**

Figure 6: Evaluation of DeepR with its variants on the Beijing and Chengdu dataset.

the above GNN models, it achieves the best performance among all the baselines. However, HAN is not capable of capturing the spatial features of POI and the aspect features well. DeepR can learn multi-context information effectively from the HPIN. Therefore, as shown in Table 1, the performance of DeepR is improved greatly, where the ACC of DeepR is higher than Geom-GCN by 5.25% and HAN by 4.52% on the Beijing dataset.

6.2.2 How Spatial Factor Helps. To study how spatial factor helps with the prediction, we design different variants of DeepR:

- **DeepR-C**: It uses the co-query graph information without heat map based CNN to learn spatial distribution.
- **DeepR-L**: Location-aware attention mechanism is removed.
- **DeepR-S**: It drops spatial oriented aggregation in SA-GNN.
- **DeepR-P**: There is no spatial-dependency attentive propagation in SA-GNN.

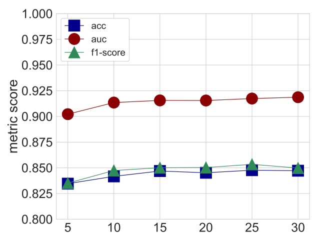
As we can see in Figure 6(a) and 6(c), DeepR can outperform all other variants of DeepR-C, DeepR-L, DeepR-S, and DeepR-P on two datasets. It demonstrates the effectiveness of our proposed model to handle the spatial information. Specifically, our proposed model outperforms DeepR-C, which shows the effectiveness of considering spatial distant-range dependencies in the heat map. The performance of DeepR-L is poor obviously on all three metrics, showing that the spatial information for attentive propagation is significant. Furthermore, if we remove the spatial oriented aggregation (DeepR-S) or spatial-dependency attentive propagation completely (DeepR-P), results get worse greatly, which indicates the necessity of the SA-GNN to overcome the limitations of the general message-passing GNNs in the POI graph learning problem.

6.2.3 How Aspect Context Helps. We further study how the aspect context can help to predict the competitive relationship effectively. We conduct experiments on variants:

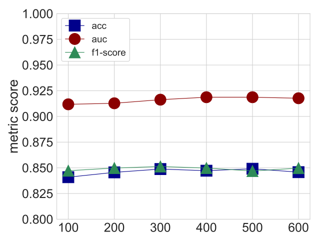
- **DeepR-CA**: Cross attention layer of PKE is removed.
- **DeepR-A**:The PKE component in DeepR is removed.

Figure 6(b) and 6(d) show that removing aspect context learning component results in the accuracy decreased by 2.6%, which justifies the importance of aspect information. Moreover, DeepR-CA gets worse performance than the proposed model, because noisy aspects can disturb prediction results without cross attention.

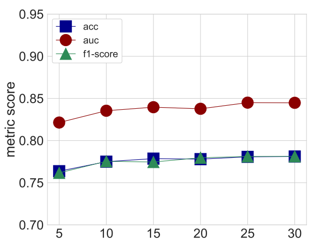
6.2.4 Hyper-parameters Sensitivity. We also conduct two experiments to study how parameter k and sl influence DeepR's performance. As illustrated in Figure 7, when the number of aspects k increases from 0 to 20, there are noticeable improvements on all metrics. But when $k \geq 10$, results don't change much. It is because that the top aspects (i.e., top 10) sorted by TF-IDF are more important and the low-ranking aspects are relatively less semantic for prediction. Figure 7 shows the scores increase slowly at the beginning and become relatively stable when the size of the grid is larger than $400m \times 400m$. With the increase of the grid size, on the one hand, POIs in a grid can capture more distant Euclidean spatial information, while on the other hand, this information from the bigger granularity is not always good. As a result, the performance achieves an equilibrium when $sl \geq 400$.



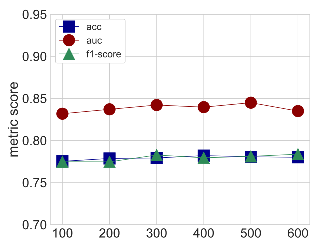
(a) Effect of topk aspects on **Beijing**



(b) Effect of grid size on **Beijing**



(c) Effect of topk aspects on **Chengdu**



(d) Effect of grid size on **Chengdu**

Figure 7: Sensitivity analysis on two datasets.

7 CONCLUSION

In this paper, we study the competitive analysis problem for POIs which is valuable on the commercial front. We construct a heterogeneous POI information network and propose the DeepR framework to discover the competitors of POIs. Moreover, spatial adaptive graph neural network is designed in DeepR, aiming at overcoming the limitations of message-passing GNNs. Meanwhile, the proposed model utilizes the pairwise POI knowledge extracted from reviews to improve the performance. Extensive experimental results on two real-world datasets show that DeepR significantly outperforms all baselines for POI competitive relationship prediction.

ACKNOWLEDGMENTS

This research was partially supported by grants from the National Key Research and Development Program of China (Grant No.2018YFB1402600), and the National Natural Science Foundation of China (Grant No.91746301,71531001, 61725205,61703386, U1605251).

REFERENCES

[1] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.
[2] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.
[3] Mahdi Jalili, Yasin Orouskhani, Milad Asgari, Nazanin Alipourfard, and Matjaž Perc. 2017. Link prediction in multiplex online social networks. *Royal Society open science* 4, 2 (2017), 160863.
[4] Nitin Jindal and Bing Liu. 2006. Identifying comparative sentences in text documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 244–251.
[5] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[6] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[7] Rui Li, Shenghua Bao, Jin Wang, Yong Yu, and Yunbo Cao. 2006. Cominer: An effective algorithm for mining competitors from the web. In *Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 948–952.
[8] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology* 58, 7 (2007), 1019–1031.
[9] Hao Liu, Ting Li, Renjun Hu, Yanjie Fu, Jingjing Gu, and Hui Xiong. 2019. Joint Representation Learning for Multi-Modal Transportation Recommendation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. 1036–1043.
[10] Hao Liu, Yongxin Tong, Panpan Zhang, Xinjiang Lu, Jianguo Duan, and Hui Xiong. 2019. Hydra: A Personalized and Context-Aware Multi-Modal Transportation Recommendation System. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2314–2324.
[11] Weiping Liu and Linyuan Lü. 2010. Link prediction based on local random walk. *EPL (Europhysics Letters)* 89, 5 (2010), 58007.
[12] Zhongming Ma, Gautam Pant, and Olivia RL Sheng. 2011. Mining competitor relationships from online news: A network-based approach. *Electronic Commerce Research and Applications* 10, 4 (2011), 418–427.
[13] Xing Niu, Xinruo Sun, Haofen Wang, Shu Rong, Guilin Qi, and Yong Yu. 2011. Zhishi. me-weaving chinese linking open data. In *International Semantic Web Conference*. Springer, 205–220.
[14] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.
[15] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.
[16] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 593–607.
[17] George Valkanas, Theodoros Lappas, and Dimitrios Gunopulos. 2017. Mining Competitors from Large Unstructured Datasets. *IEEE Transactions on Knowledge and Data Engineering* 29, 9 (2017), 1971–1984.
[18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[19] Hao Wang, Tong Xu, Qi Liu, Defu Lian, Enhong Chen, Dongfang Du, Han Wu, and Wen Su. 2019. MCNE: An End-to-End Framework for Learning Multiple Conditional Network Representations of Social Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1064–1072.
[20] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. 2018. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 592–600.
[21] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. 2015. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences* 58, 1 (2015), 1–38.
[22] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous Graph Attention Network. In *The World Wide Web Conference*. ACM, 2022–2032.
[23] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. 2019. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596* (2019).
[24] Jin Xu, Jingbo Zhou, Yongpo Jia, Jian Li, and Xiong Hui. 2020. An Adaptive Master-Slave Regularized Model for Unexpected Revenue Prediction Enhanced with Alternative Data. In *IEEE 36th International Conference on Data Engineering*. IEEE, 601–612.
[25] Kaiquan Xu, Stephen Shaoyi Liao, Jiexun Li, and Yuxia Song. 2011. Mining comparative opinions from customer reviews for competitive intelligence. *Decision support systems* 50, 4 (2011), 743–754.
[26] Yang Yang, Jie Tang, Jacklyne Keomany, Yanting Zhao, Juanzi Li, Ying Ding, Tian Li, and Liangwei Wang. 2012. Mining competitive relationships by learning across heterogeneous networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 1432–1441.
[27] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7370–7377.
[28] Zikai Yin, Tong Xu, Hengshu Zhu, Chen Zhu, Enhong Chen, and Hui Xiong. 2019. Matching of social events and users: a two-way selection perspective. *World Wide Web* (2019), 1–19.
[29] Le Zhang, Tong Xu, Hengshu Zhu, Chuan Qin, Qingxin Meng, Hui Xiong, and Enhong Chen. 2020. Large-Scale Talent Flow Embedding for Company Competitive Analysis. In *Proceedings of The Web Conference 2020*. 2354–2364.
[30] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*. 5165–5175.
[31] Weijia Zhang, Hao Liu, Yanchi Liu, Jingbo Zhou, and Hui Xiong. 2020. Semi-Supervised Hierarchical Recurrent Graph Neural Network for City-Wide Parking Availability Prediction.
[32] Jingbo Zhou, Shan Gou, Renjun Hu, Dongxiang Zhang, Jin Xu, Airong Jiang, Ying Li, and Hui Xiong. 2019. A Collaborative Learning Framework to Tag Refinement for Points of Interest. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1752–1761.

A APPENDIX

A.1 Additional Dataset Description

Here we introduce how to construct ground truth. Table 2 shows the instruction of the main attributes of POI we used in our work. At first, for each brand, we collected its related brand (i.e. KFC and Mcdonald's) from a public knowledge base zhishi.me [13] which has a "relatedPage" relation. For all POI pairs between two related brands, we use the following conditions to pick up POI pairs with competitive relationship: 1) The distance between two POIs should be within 10 km;) They have the same category; 3) The overlap of the checked-in users of two POIs in July 2018 is larger than 5%.

In total, the Beijing dataset contains 18,731 pairs and the Chengdu dataset contains 7,514 pairs. We manually check 200 pairs randomly selected from the ground truth, and find that the accuracy is larger than 95%. We generate the same number of negative POI pairs with the same category for each city by randomly sampling while also limiting their distances within 10 km, and we also control that the distribution of POIs in each category is consistent with the distribution of positive samples.

A.2 Baseline Description

We compare our DeepR model with the following methods to predict the competitive relationship of POIs:

- **DIST** (distance rule) and **EW** (edge weight rule) are two simple methods. We set the best threshold to determine whether POI pairs are competitive or not. For these baselines, if the distance (w.r.t. **DIST**) or edge weight (w.r.t **EW**) of the POI pairs with the same category is smaller than a pre-defined threshold, they are considered to be competitive.
- **MLP** and **XGBoost**[1]. We concatenate POI features (i.e., category and coordinate) of a pair of POIs and correlation features (i.e., distance and co-query weight) as input to predict competitive relationship by MLP and XGBoost.
- **Deepwalk**[15] and **Node2vec**[2]. We use graph embedding methods to learn POI's embedding on co-query graph, and then apply an MLP classifier to predict.
- **GCN**[6]. It is a kind of graph neural network, which aggregates node features on graph. Here we use POI features and the average vector of aspect embeddings learned from word2vec as the input feature of each node, and GCN is conducted on co-query graph.
- **GAT**[18]. This model is also a graph neural network considering the attention mechanism to learn appropriate weights of neighbor nodes. Here the input is the same as GCN.
- **Geom-GCN**[14]. Geom-GCN is based on a geometric aggregation scheme for graph neural networks. Here we build the model on the co-query POI graph, and the input feature is the same as GCN.
- **SEAL**[30]. It is a state-of-art graph neural network for link prediction. Here we feed it with the subgraph extracted from the link, and the input feature has three components: structural POI labels, POI embeddings and POI attributes.
- **HAN**[22]. HAN is a state-of-art heterogeneous graph neural network, which employs node-level attention and semantic-level attention. Here we construct three meta-paths (PP, PBP, PBABP) for HAN to learn POI representations to predict the competitive relationship.

Table 2: The main attributes of POI we used in our work.

Attribute	Description
Name	The name of POI.
Category	Standard two-level category.
Brand	Brand information of POI.
Point x	The x-coordinate of POI on the map.
Point y	The y-coordinate of POI on the map.

A.3 Experiment Setup

A.3.1 Data Splitting. In the experiment, we randomly pick 10% of data for the test and 10% of data as the validation set while limiting their corresponding brand pairs not appearing in the remaining 80% train set in order to avoid information leakage.

A.3.2 Parameter Settings. For DeepR, we set the embedding size of each node in HPIN as 100, and the size of every hidden layer the same as embedding size. We set $\Delta t = 30$ min, $\theta_m = 50$ and $\theta_{PMI} = 0.2$. We select the top 30 aspects of each brand and the window size is 5. We divide the neighbors of each POI into four sectors for the spatial oriented aggregation of SA-GNN. For spatial-dependency attentive propagation, we set the bucket as 100 meters, and the maximum distance in each dimension is 10km. We set the grid size in the heat map as 500m×500m, set $L = 11$ and select 12 categories as channels, which means $C = 12$. In addition, we set the learning rate to 0.01, the filter size of CNN to 2×2 and 3×3, the number of attention head to 8, the dropout to 0.5, the L_2 loss weight to 1e-5. We use Adam [5] as the optimizing method and Relu as the activation function $\sigma(\cdot)$

For baseline models, we tune the parameters of each model to ensure the best performance. More specifically, for the rule-based methods, the threshold of distance for DIST is 4.2 km, and the threshold of co-query edge weight for EW is 227, meaning that two POIs are queried 227 times by common users. For feature-based methods, we concatenate POI features of a pair of POIs and correlation features as input to train MLP and XGboost. We adopt three-layer structure for MLP. The numer of decision tree in XGboost is set to 200 and the max-depth of trees is set to 5. For fair comparison, the embedding dimension d of all other baselines are set to 100 (same as DeepR). For the graph embedding methods (Deepwalk and Node2vec), we set the walk length as 10 and the number of random walk as 80. The window size is set to 5, and the parameter q and p for Node2vec are both set to 1. For GCN, the input includes POI features and the average vector of aspect embeddings learned from word2vec as the input feature of each node, and GCN is conducted on co-query graph. For SEAL, we use the output of Node2vec as the embedding part of the node features. For GAT and Geom-GCN, the input is the same as the GCN. The number of heads and the numbers of hidden units for GAT are set to 8 and 16, respectively. We use the Struc2vec as the graph embedding method to learn the latent space feature for Geom-GCN. For HAN, we employ three meta-paths, i.e., PP (POI-POI), PBABP (POI-brand-aspect-brand-POI) and PBP (POI-brand-POI) to train the model.