Noname manuscript No.
(will be inserted by the editor)

Semantic Embedding for Regions of Interest

Debjyoti Paul, Feifei Li, Jeff M. Phillips
School of Computing, University of Utah
{deb, lifeifei, jeffp}@cs.utah.edu

the date of receipt and acceptance should be inserted later

Abstract The available spatial data is rapidly growing and also diversifying. One may obtain in large quantities information such as annotated point/place of interest (POIs), check-in comments on those POIs, geo-tagged microblog comments, and demarked regions of interest (ROI). All sources interplay with each other, and together build a more complete picture of the spatial and social dynamics at play in a region. However, building a single fused representation of these data entries has been mainly rudimentary, such as allowing spatial joins. In this paper, we extend the concept of semantic embedding for POIs (points of interests) and devise the first semantic embedding of ROIs, and in particular ones that captures both its spatial and its semantic components. To accomplish this, we develop a multipart network model capturing the relationships between the diverse components, and through random-walk-based approaches, use this to embed the ROIs. We demonstrate the effectiveness of this embedding at simultaneously capturing both the spatial and semantic relationships between ROIs through extensive experiments. Applications like popularity region prediction demonstrates the benefit of using ROI embedding as features in comparison with baselines.

1 Introduction

In the last decade, location-based social networks (LBSNs) like Facebook, Instagram, Foursquare, Twitter have attracted billions of users, where people can check in at point of interests (POIs) and share life experience in the physical world via mobile device promptly. It is crucial for such service providers to leverage the data they collected to make personalized recommendations that help their users to explore new places and facilitate targeting advertisement for generating revenue [3,8,26]. Recent literature suggests that distributed representation of point of interest (POI) or embedding can further improve the results [18,43,47,56]. It is worth to note that point of interest (POI) are a single point/place on the map of Earth (e.g. *New York Stock Exchange, New York*).

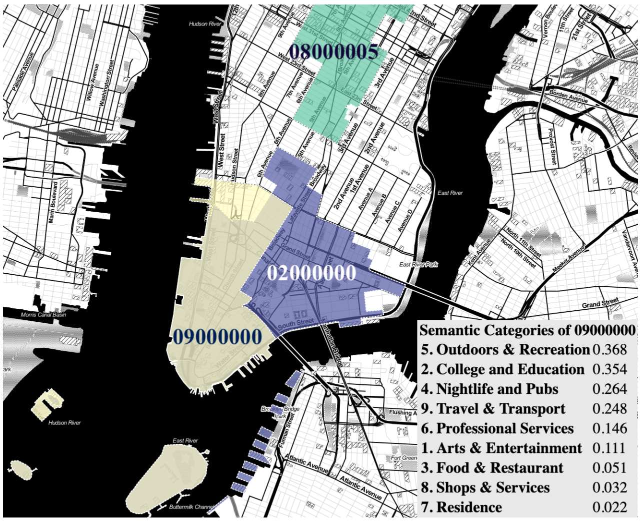

08000005
02000000
09000000
Semantic Categories of 09000000
5. Outdoors & Recreation 0.368
2. College and Education 0.354
4. Nightlife and Pubs 0.264
9. Travel & Transport 0.248
6. Professional Services 0.146
1. Arts & Entertainment 0.111
3. Food & Restaurant 0.051
8. Shops & Services 0.032
7. Residence 0.022
Hudson River
East River
Grand Street
Atlantic Avenue

Fig. 1: ROI examples from Lower Manhattan, New York.

Recently, an increasing interest on studying *region of interest* (ROI) [45] is rising [39], where the social dynamics occurring at POIs located in a particular region is considered as a whole. By picturing the semantic and spatial features of different regions intertwined with people's activities can yield important information such as functional behavior,

Address(es) of author(s) should be given distinctive features and social effects, which can be further utilized in urban planning and region-level recommendation.

An example of the application is shown in Figure 1; ROIs 02000000 (blue), 08000005 (green) are semantically as well as spatially correlated with ROI 09000000 (yellow) from *Manhattan, New York City.* Semantic category information of ROI 09000000 is also presented in Figure 1 where *Outdoors & Recreation, College and Education, Nightlife and Pubs, Travel and Transport, Professional Services* are presented as top five major categories based on cosine similarity metric. A careful observation in the map will reveal that ROI 09000000 consists of Statue of Liberty, Ellis Island & Battery Park and World Trade Center which has been visited by more than 3.5 million visitors in an average for the last 5 years [30], is a major reason for *Outdoors & Recreation* as the topmost category. New York University, The King's College and Pace University etc. are also demarked within the region that follows the second top spot as *College and Education.* The next three top categories are intuitive to estimate since Lower Manhattan is the hub of some popular old pubs, financial offices, hotels and well connected subway, transport & ferry system in the city. Though ROI 08000005 is geospatially distant from ROI 09000000 but they are semantically similar in terms of *Outdoor & Recreation, College & Education* and *Travel & Transport* because of Central Park, New York University Midtown Campus, Pace University, Grand Central Terminal and major subway connections respectively.

An effective approach to capture both semantic and spatial feature at the same time is to *embed them in a latent semantic space* as elaborated by [43, 47, 48] for POIs. Hence, embedding over ROI with semantic features would also be an effective method for ROI analysis. Nevertheless, existing solutions only consider the semantic embedding of POI but not ROI. A naive extension for extending semantic embedding for POI to obtain semantic embedding of ROI is to simply aggregate over POI features for all POIs inside a ROI and treat that ROI then simply as an aggregated POI. However, this approach is not effective in capturing spatial and semantic information simultaneously due to the loss of interesting correlations between spatial and semantic information in the process of simple aggregation, as also verified by our experiments. We deduced ROI embedding problem into a tripartite graph embedding problem with entities (a) ROIs (b) POIs (c) Words, whose embedding goal is to minimize the probability distribution difference between embedding entities in latent space and the information graph network based on edge connections. The ROI embedding model facilitates the online analysis and discovery of the (dis)similarities between any pair of ROIs from the perspective of human understanding.

To further add to our motivation, and answer why ROI embedding is needed, we need to look at the advantages of using embedding over raw information or semantic keyword based search. Firstly, computation efficient embeddings are generic and aggregate latent features that can easily be integrated into downstream tasks. Secondly, to comply with data retention policies and maintain security standards, it is essential to limit raw information access and step towards a generic and lossy embedding. Thirdly, semantic ROI embeddings grant measurable techniques to attribute a region and can account for its change over time.

Application-wise, incorporating ROI embedding as a feature for ad services can have a significant impact, as localized crowd engagement/activity in neighborhoods can promote economic growth. ROI features is another step towards improvement of localized search results. Another far reaching application of ROI using features is vacation home rentals recommendations based on user's neighborhood preferences. Semantic embedding of ROIs also enables users to filter with scores on each categories like *Travel & Transport, Shops & Services, Arts & Entertainment, Schools* or *Nightlife* for finding listings with neighborhood information.

The main set of challenges of ROI semantic embedding comparing against POI semantic embedding lies in:

1) Geographic Influence: Recent studies on POI embedding can effectively classify the POIs categories and use them as features for prediction and recommendation applications. However, evaluating the influence of POIs on its neighborhood region is challenging and not yet been addressed in literature.

2) Capturing Social Effect: Social responses from microblog sites are highly dynamic and often captures popularity information of places in any region. Discovering any spatial features from social behavior is complex and involves significant challenges.

3) Semantic challenges: Leveraging the textual information associated with places and regions to obtain semantic features is a non-trivial task. We modeled a tripartite graph network embedding approach to learn ROI embedding.

4) Data Challenges: It is difficult to get a large and open dataset of POIs with textual information from location-based social networks. Currently available public datasets are either geographically sparse or not suitable for our problem statement. We resort to scraping and crawling for creating appropriate datasets for our investigation.

We summarize the contributions of this paper.

- We formulate the Region of Interest (ROI) semantic embedding that *simultaneously embeds into semantic space and spatial space.* That is, this embeds ROIs nearby others in latent space, which are both nearby semantically in terms of the dominant places of interest as well as the social discussion within those regions, and nearby geospatially, in that way they are nearby other neighboring and overlapping ROIs.
- We propose a Tripartite Network Embedding (TNE) to learn a low-dimensional representation of ROIs. For property preserving embedding, TNE introduces, *(a) Mergeable Indirect Graphs:* TNE creates transitive relation preserving more informative homogeneous graphs, then proposes a method for compatibility testing to merge multiple homogeneous graphs. *(b) Community-aware Random Walks:* TNE alleviates moderately connected community problem in graphs. *(c) Heterogeneous Negative Sampling:* TNE proposes noise distributions on heterogeneous graphs which enhances learnability. TNE is easily extendable to multi-partite network embedding problems.
- We introduce a semantic category annotation for ROIs to identify the feature similarities of ROI with defined categories for semantic understanding. It also helps us to evaluate ROI embeddings in our experiments.
- We present extensive experiments with real-world datasets to show qualitative advantage of ROI embedding with TNE. We compared TNE with state-of-the-art baselines to justify our embedding process through spatial and semantic facets.

The rest of the paper is organized as follows. Section 2 presents our problem formulation with baseline approaches inspired by state-of-the-art literature. In Section 3, we present our model TNE: Tripartite Network Embedding followed by experiments in Section 4, related works in Section 5 and Section 6 concludes the paper.

2 Preliminaries

This section introduces problem formulation with some necessary definitions and notations used in the paper. After that, we present problem statements on semantic ROI embedding formally and describe our information graph network. Next, we enlist a few baseline approaches to compare with our tripartite network embedding model, TNE.

2.1 Problem Formulation

Assume we have three sets of data: points of interest (POIs), regions of interest (ROIs) and geotagged documents. We define each next.

A *Region of Interest (ROI)* dubbed as r is an area in the map of Earth demarked by a geometry of circle, rectangle or polygon, e.g., ROI 09000000 from Figure 1. An ROI r=*(id, geofeatures, name, properties)* is a tuple of identifier, polygonal geofeatures, name and optional properties like state, country respectively. ROIs are technically stored as GeoJSON [17]. We represents a set of ROIs as $R = \{r_1, r_2, \ldots, r_{|R|}\}$.

A *Point of Interest (POI)* dubbed as p is defined as a specific point location in the map of Earth. E.g. Empire State Building, New York. A POI p=*(id, coord, name, properties)* is a tuple of identifier, latitude-longitude geo-coordinate, name and optional properties like keywords, description, address and category respectively. It is also stored as GeoJSON object. A set of POIs is represented as $P = \{p_1, p_2, \ldots, p_{|P|}\}$.

A *geotagged document* dubbed as d is a geolocation associated textual record either by origin or reference. E.g., check-in comments, reviews, microblogs etc. A geotagged document d = *(id, text, coord, properties)* is a tuple of identifier, text, a latitude-longitude geo-coordinate and optional properties like timestamp, user information. We are mainly interested in two types of geotagged documents *(a) Microblogs (b) Social reviews.* Microblog documents associated with a ROI r is denoted as D_r, and social review documents associated with a POI p is denoted as D_p. Documents are associated with POIs and ROIs based on geotagged locations. We define all geotagged documents as $D = D_R \cup D_P$ where D_P and D_R are, $D_P = \bigcup_{p\in P} D_p$; $D_R = \bigcup_{r\in R} D_r$. Words from D forms vocabulary set $W = \{w : w \in d.text, d \in D\}$.

We capture relations among multiple entities (i.e., POI, ROI, and Words) through the information graph described in Section 2.3. In Table 2, we summarize all the notations used in this paper.

2.2 Problem Statements

Problem 1 (Semantic Embedding of ROI) Given a set of ROIs R, a set of POIs P, an associated set of geotagged documents D and an embedding dimension n, the goal of semantics embedding of ROI is to embed each ROI $r \in R$ as a vector $\vec{\boldsymbol{r}} \in \mathbb{R}^n$, such that the cosine distance of $\vec{\boldsymbol{r_i}}$ and $\vec{\boldsymbol{r_j}}$ captures the similarity of r_i and r_j in both spatial and semantic aspects.

Table 1: Table of Notation

Notation	**Meaning**
$R = \{r_1, r_2, r_3, \dots, r_{\|R\|}\}$	A set of ROIs.
$P = \{p_1, p_2, p_3, \dots, p_{\|R\|}\}$	A set of POIs.
$D_r = \{d_1, d_2, d_3, \dots, d_{\|D_r\|}\}$	A set of geotagged documents associated with ROI r.
$D_p = \{d_{\|D_r\|+1}, d_{\|D_r\|+2}, \dots, d_{\|D_r+D_p\|}\}$	A set of geotagged documents associated to POI p.
$D = D_R \cup D_P$	A set geotagged documents.
$W = \{w_1, w_2, \dots, w_{\|W\|}\}$	Vocabulary set or Words.
$d.text = \{w_1, w_2, \dots : w_i \in W\}$	Text of document d.
G_{rp}	ROI-POI Bipartite Graph.
G_{rw}	ROI-Words Bipartite Graph.
G_{pw}	POI-Words Bipartite Graph.
G_r	ROI Graph.
G_w	Word Graph.
$\mathcal{G} = (G_{rp}, G_{rw}, G_{pw}, G_r, G_w)$	Information Graph Network.
$\vec{\boldsymbol{r}}$	Embedding of ROI r.
$\vec{\boldsymbol{p}}$	Embedding of POI p.
$\vec{\boldsymbol{w}}$	Embedding of Word w.
$C = \{c_1, \dots, c_{\|C\|}\}$	Semantic Categories.
$\mathcal{C} = \{\vec{c_1}, \dots, \vec{c_{\|\mathcal{C}\|}}\}$	The set of semantic category vectors.

The objective of ROI semantic embedding is capturing geographic information and the semantic perspective from the crowd about the region. If any ROI stands out in any semantic features such as recreational activities, office & services, residential region or any combination of activities, that must be captured via embedding. We introduce an application of ROI embedding as Problem 2: Semantic Category Annotation for evaluation of ROI embedding.

Problem 2 (Semantic Category Annotation of ROI) Given a semantic embedding $\vec{\boldsymbol{r}}$ for ROI r and a set of categories $C = \{c_1, \dots, c_{|C|}\}$, where a category $c_i = \{w_1, w_2, \dots\}$ is represented with set of words, the goal is to annotate the ROI r with semantic categories $\boldsymbol{Sem_r} = \{c_i : score_i | \forall c_i \in \mathcal{C}\}$ with corresponding similarity scores.

The aim of Problem 2 is to semantically annotate any ROI r from the generated ROI semantic embedding $\vec{\boldsymbol{r}}$. As we know, word representation in semantic space are capable of capturing its meaning via context or synonyms in close proximity space. Firstly, we propose a systematic approach to define semantic category which adheres to the categories defined in Table 2. In our model we describe a category c with a set of words $\{w_1, w_2, \dots, w_k\}$ that captures meaningful information about that category. For example, the category *Travel and Transport* is described with words *travel, trip, station, train, ferry, car, airport, pier* etc. We perform a normalized average of these word vectors (each word is represented by a vector via an word embedding process, e.g., Word2Vec embedding) to represent the vector for the semantic category which we dub as *semantic category vector* $\vec{\boldsymbol{c}}$. The cosine similarity score of a *semantic category vector* $\vec{\boldsymbol{c}}$ with an ROI $\vec{\boldsymbol{r}}$ i.e. normalized dot product $\langle \vec{\boldsymbol{r}}, \vec{\boldsymbol{c}} \rangle$ determines the closeness of ROI with respective semantic category. The goal of this study is to find how well we can annotate an ROI with sentiment categories $\mathcal{C} = \{\vec{c_1}, \dots, \vec{c_9}\}$ and whether it adheres to real world scenarios. An example of ROI semantic category annotation is given in bottom corner of Figure 1.

Table 2: A set of Sampled Categories $\mathcal{C}$

🟦	1. Arts & Entertainment	🟦	2. College & Education
🟦	3. Food	🟩	4. Nightlife Spot
🟩	5. Outdoors & Recreation	🟩	6. Professional & Other Places
🟨	7. Residence	🟧	8. Shop & Service
🟥	9. Travel & Transport		

2.3 Information Graph Network

We define an *information graph network* $\mathcal{G} = (G_{rp}, G_{rw}, G_{pw}, G_r, G_w)$, which is a combination of graphs with POI, ROI and Word entities to capture spatial and semantic information, illustrated in Figure 2. It is to note that vocabulary of semantic words $W = \{w : w \in d.text, d \in D\}$ is from geotagged documents.

In our model, the information graph $\mathcal{G}$ is formed of multiple subgraphs. The subgraphs we model are of two types *heterogeneous or bipartite* subgraphs and *homogeneous* subgraphs. We define the three bipartite subgraphs and two homogeneous subgraphs as follows.

Definition 1 (ROI-POI Bipartite Graph: G_{rp}) An ROI-POI graph, denoted as $G_{rp} = (R \cup P, E_{rp})$, is

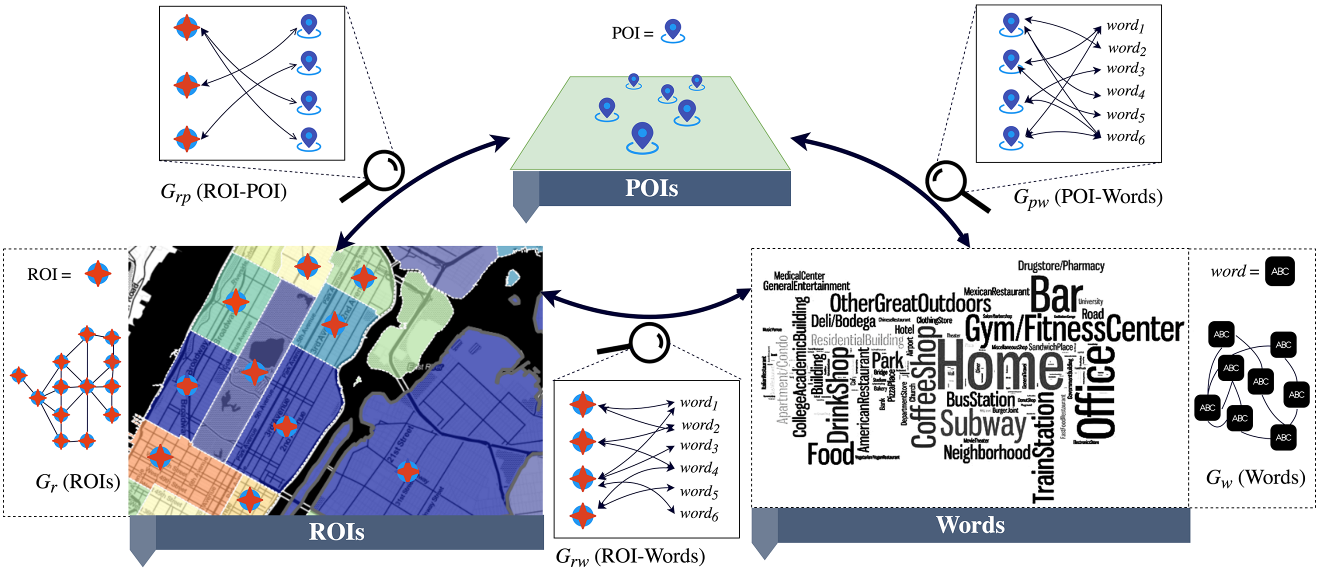


Fig. 2: Information Graph Network $\mathcal{G}$ with illustration of ROI-POI bipartite graph G_{rp}, ROI-Word bipartite graph G_{rw}, POI-Word bipartite graph G_{pw}, ROI graph G_r and Word graph G_w.

a bipartite graph with edges E_{rp}. An edge $\{e = (r_i, p_j) \in E_{rp}\}$ exists iff p_j is located within r_i and the weight of edge is $\omega(r_i, p_j) = 1$.

Definition 2 (ROI-Word Bipartite Graph: G_{rw}) An ROI-Word graph, denoted as $G_{rw} = (R \cup W, E_{rw})$, is a bipartite graph with edges E_{rw}. An edge $\{e = (r_i, w_j) \in E_{rw}\}$ exists iff w_j is mentioned in any d_{r_i} and the weight of edge $\varpi(r_i, w_j)$ is calculated with tf-idf scores.

Definition 3 (POI-Word Bipartite Graph: G_{pw}) A POI-Word graph, denoted as $G_{pw} = (P \cup W, E_{pw})$, is a bipartite graph with edges E_{pw}. An edge $\{e = (p_i, w_j) \in E_{pw}\}$ exists iff w_j is mentioned in any d_{p_i} and the weight of edge $\varpi(p_i, w_j)$ is calculated with tf-idf scores.

Definition 4 (ROI Graph: G_r) An ROI graph, denoted as $G_r = (R, E_r)$, is a homogeneous graph network of ROIs where an edge $e \in E_r$ between two ROIs denote they are spatially overlapped or neighboring region.

Definition 5 (Word Graph: G_w) An word graph, denoted as $G_w = (W, E_w)$, is a homogeneous graph network of words where an edge $e \in E_w$ between two words signifies their co-occurrence in geotagged documents.

2.4 Baseline Approaches

State-of-the-art methods are not tailored for multi-partite embedding. Hence, we extend the state-of-the-art network embedding methods to form multiple comparable baselines to compare against our TNE model with various experiments in Section 4. The last baseline TNE_NW use TNE model but trained on non-weighted edges version of information graph. We use this baseline to show the importance of edge weights in our model.

1. GE_POI *(POI Aggregation):* The work of Xie et al. [47] produces state-of-the-art POI embedding $\{\vec{\boldsymbol{p}} : p \in P\}$ for POI recommendation. Though their objective is different from our but we matched their technical model with our information graph network $\mathcal{G} \setminus (G_{rw}, G_r)$ to generate POI embedding for fair comparison. Based on the edges of graph G_{rp} we aggregate POI embedding $\vec{\boldsymbol{p}}$ from same ROIs via normalized vector summation to obtain resultant ROI embedding vector $\vec{\boldsymbol{r}}$. This baseline also depicts what if similar importance is given to all the POIs in a region. Mathematically $\vec{\boldsymbol{r}}$ is calculated as:

$$\vec{\boldsymbol{r}} = \left\{ \frac{\sum \vec{\boldsymbol{p}}}{||\sum \vec{\boldsymbol{p}}||} : \exists (r, p) \in E_{rp}, r \in R \right\}$$

This approach is expected to perform well in POI embedding and capturing basic (non-weighted) semantic relation of ROIs. Since, GE_POI does not perform crowd engagement within ROIs, it is expected to partially solve the problem.

2. CrossMap *(Region-Word graphs):* This approach is very similar to CrossMap work from Zhang et al. [54] for popular event exploration. Inspired by their model, we only leverage the information from social engagements and its relation on ROIs to generate ROI embedding. From our information graph $\mathcal{G}$, we used G_{rw}, G_r, G_w to generate ROI embedding for this baseline. Baseline CrossMap captures crowd engagements on ROI but neither the geospatial correlation nor the POI effects is considered for ROI in their approach.

3. BiNE *(Multiple Bipartite Networks):* BiNE [16] is a method proposed for learning vertex representation in bipartite graphs. We treat this as another baseline for learning ROI embedding from bipartite graphs G_{rp}, G_{pw}, and G_{rw}. It will be interesting to see if this baseline can capture the spatial affinity and semantic relation of ROIs. We expect BiNE to fail in capturing geospatial correlation as transitivity property is not incorporated in this approach. In the related work (i.e., section 5), we explicate the rationale of using BiNE as another state-of-the-art baseline model for comparisons.

4. TNE_wcr *(TNE without Community Random Walk):* This version of our model TNE does not take advantage of our community-aware random walk strategy and uses the traditional random walk strategy. Including this baseline model in our experiments helps recognize the impact of incorporating the community-aware random walk in TNE.

5. TNE_nw *(Non-weighted TNE):* This version of our model TNE does not use *tf-idf* weights over G_{pw} graph for measuring the popularity of POIs. This approach demonstrates the modeling advantage of these weights in comparison to Jenkins et al. [22] which does not use such weights – among other differences.

3 TNE: Tripartite Network Embedding

In this section present our approach TNE, the Tripartite Graph Network representation learning which can be generalized to a multipartite network embedding model. The primary focus of TNE is learning of ROI embedding, i.e. Problem 1 and Problem 2 is an application of the former. Our network embedding model TNE is *(a) microscopic structure-preserving network embedding; (b) transitive property preserving networks; and (c) community-aware network embedding.* We explain each of these features as we simplistically unravel our model.

3.1 Direct Relation Models

The relationship among vertices which is straight-forward visible from the edges set in the information network is known as *direct relation model.* We classify *Direct Relation Models* based on type of vertices between the edges in graph, such as, *(a) Heterogeneous Relation (b) Homogeneous Relation models.*

3.1.1 Direct Heterogeneous Relation Model

The basic graph building block for any multipartite/tripartite networks are bipartite networks that represent relationships between two non-similar entities or vertices set. Consider our tripartite information network $\mathcal{G}$, we have three bipartite networks G_{rp}, G_{rw}, G_{pw}. A bipartite graph network is a heterogeneous vertex network (in our model) that represents direct or first order relations which we dub as *Direct Heterogeneous Relation Model.*

In any structure preserving network embedding, it is desirable that the closeness property between two well-connected vertex is high. Even if the connected vertices are different in nature (e.g. POI and Word in G_{pw}) but their proximity in network is a direct relational information must be imbibed in the embedding network. For the sake of understanding, let us consider a bipartite network $G_{uv} = (U \cup V, E_{uv})$ where $U = \{u_1, u_2, \ldots, u_{|U|}\}$ and $V = \{v_1, v_2, \ldots, v_{|V|}\}$ are two sets of different types of vertices, and $E_{uv} \subset U \times V$ is edge set. Also consider the embedding representation of vertex u_i and v_j as $\overrightarrow{\boldsymbol{u_i}} \in \mathbb{R}^n$ and $\overrightarrow{\boldsymbol{v_j}} \in \mathbb{R}^n$ respectively. In our model we consider the Euclidean embedding space where we define *closeness* measure between any two vertices u_i and v_j as conditional probability $\overline{\Pr}(v_j|u_i)$.

$$\overline{\Pr}(v_j|u_i) = \frac{\exp(-\overrightarrow{\boldsymbol{v_j}}^T \overrightarrow{\boldsymbol{u_i}})}{\sum_{v_k \in V} \exp(-\overrightarrow{\boldsymbol{v_k}}^T \overrightarrow{\boldsymbol{u_i}})} \tag{1}$$

Existing literatures and pioneer embedding work of word2vec [28] depicts the importance of using inner product for similarity measure and transforming it into probability space with sigmoid function. The microscopic structure of network connection is captured with conditional probability between vertices.

$$\Pr(v_j|u_i) = \frac{\varpi(u_i, v_j)}{deg_{u_i}} \tag{2}$$

where ϖ is edge weight function i.e. $\varpi(u_i, v_j)$ is weight of edge $e_{u_i v_j} \in E_{uv}$ and $deg_{u_i} = \sum_{e_{u_i v_k} \in E_{uv}} \varpi(u_i, v_k)$.

The objective of the model is to learn the embedding vectors by minimizing difference between pairwise distribution.

$$\begin{aligned} \text{minimize } O &= D_{KL}(\Pr || \overline{\Pr}) \\ &= - \sum_{e_{u_i v_j} \in E_{uv}} \Pr(v_j|u_i) \log \frac{\overline{\Pr}(v_j|u_i)}{\Pr(v_j|u_i)} \\ &= - \sum_{e_{u_i v_j} \in E_{uv}} \varpi(u_i, v_j) \cdot \log \overline{\Pr}(v_j|u_i) \end{aligned} \tag{3}$$

where D_{KL} is KL divergence measure for difference between probability distributions. The expression $-\sum \Pr(v_j|u_i) \log \Pr(v_j|u_i)$ from Equation 3 is the information entropy expression which is modeled as edge entropy i.e. $\sum \varpi(u_i, v_j)$ function. From the final expression, we obtain all the variables in optimization functions i.e. vectors $\vec{\boldsymbol{u_i}}, \vec{\boldsymbol{v_j}}$ from $\overline{Pr}(\cdot)$.

KL divergence is a particular case of a broader class of divergences called f-divergences. KL divergence is asymmetric and commonly used by embedding methods that preserves local and microstructures [21]. There are other types of divergences such as Reverse KL divergence (RKL), Jenson-Shannon (JS) divergence, Hellinger distance [19], χ^2 distance measures. As the name suggests optimizing with Reverse KL measures can capture the global or macro network structures. JS divergence is symmetric in nature, and some research works suggests using JS distance as a cost function in the empirical domain for optimization purpose [15,27]. χ^2 distance also behaves similar with respect to preserving local structure. Based on the intention of capturing micro- and macro-structure or giving equal importance to both of them, we can pick out the right methods.

In our case, the optimization equation for tripartite graphs G_{rp}, G_{rw} & G_{pw} with KL-divergence method follows:

$$\begin{aligned} \text{minimize } O_{rp} &= - \sum_{e_{r_i p_j} \in E_{rp}} \varpi(r_i, p_j) \cdot \log \overline{\Pr}(p_j|r_i) \\ \text{minimize } O_{rw} &= - \sum_{e_{r_i w_j} \in E_{rw}} \varpi(r_i, w_j) \cdot \log \overline{\Pr}(w_j|r_i) \\ \text{minimize } O_{pw} &= - \sum_{e_{p_i w_j} \in E_{pw}} \varpi(p_i, w_j) \cdot \log \overline{\Pr}(w_j|p_i) \end{aligned} \tag{4}$$

3.1.2 Direct Homogeneous Relation Model

In many information network having a direct homogeneous graph is not common. For example, consider the information network with Yahoo Answers or Quora. Users in these sites post questions which then gets answered by other users. There are direct relational graphs between *users-questions, questions-answers* and *answers-users* but there are no direct relations among *users*. There are of course information networks where direct homogeneous graphs are present. It is important that we utilize the information from such graphs because more information helps in learning better [10, 41] as it reduces uncertainty in learning weights within model.

In our scenario, G_r and G_w are two homogeneous graph in $\mathcal{G}$ i.e. the edges are between the same type of vertices. The edges in these graphs signify explicit proximity between connected vertices. Even though the information from these explicit relations are very informative but not sufficient for embedding because of their sparse nature. The embedding model can still be significantly enhanced by incorporating implicit information via indirect relation graphs as discussed in Section 3.2 and then merging direct & indirect homogeneous graphs as shown in Section 3.2.2.

3.2 Indirect Relation Models

In this section, we focus on modeling indirect and deducible relations that contributes in obtaining meaningful information towards embedding. Recent work suggests deducible information helps in improving semantic properties [16, 23, 51]. Heterogeneous networks consisting of bipartite graphs do not have explicit relations among vertices of the same type. To understand the importance of indirect relation, take the example of POIs in our data. The POIs set P does not have any explicit edges between any two POIs. But there are POIs that are similar based on the reception they receive from people. A subset of words can form a topic and commonly describe similar POIs (which is true in real world scenario), it is very likely that there will be significant number of paths between the similar POIs in bipartite graph G_{pw}. Generating all the paths between all pairs of large number of vertices is infeasible. To alleviate the issue, it is a common practice to generate several random walks to mimic the representation of a corpus of vertices with the intuition that important vertices gets repeated based on its popularity.

3.2.1 Indirect Homogeneous Graphs

Random walks on bipartite graphs have periodicity issues [1]. The common strategy of addressing this problem is to construct two homogeneous graphs from bipartite graph utilizing 2nd-order proximity between vertices of same types [11]. Having said that, we construct $G_u^v = (U, E_u^v)$, a homogeneous graph with vertices U by utilizing transitive relations with vertices V from bipartite graph G_{uv}. We defined the 2nd-order proximity between two vertices u_i and u_j by weight $\varpi(u_i, u_j)$ where $e_{u_i u_j} \in E_u^v$ such that there exists edges $e_{u_i v_k}$ and $e_{v_k u_j}$ in G_{uv}.

$$\varpi(u_i, u_j) = \sum_{v_k \in V} \varpi(u_i, v_k) \cdot \varpi(v_k, u_j) \tag{5}$$

Similarly we construct homogeneous graph $G_v^u = (V, E_v^u)$ with relations via vertices U from G_{uv}.

3.2.2 Merging Homogeneous Graphs

Our information network $\mathcal{G}$ consists of three bipartite graphs G_{rp}, G_{rw}, G_{pr}. We now generate homogeneous graph G_r^p on ROIs R with indirect relations via POIs P and homogeneous graph G_r^w on ROIs R with indirect relations via words W. Similarly, homogeneous graphs G_p^r, G_p^w are obtained on POIs P with indirect relations on ROIs R and Words W respectively. Homogeneous graphs G_w^r, G_w^p are also generated with indirect relations on ROIs R and POIs P respectively. All-inclusive G_r^p, G_r^w,G_p^r, G_p^w, G_w^r, G_w^p indirect homogeneous graphs are obtained from three bipartite graphs.

The homogeneous graphs G_r^p, G_r^w both on ROIs R provides implicit relation among its vertices. We use all the information from direct and indirect homogeneous graphs by simply appending the edges from the graphs G_r^p, G_r^w, G_r to form a single graph G_r' for modeling random walks. However, it should be determined whether these graphs are compatible and not contrasting to each other. Intuitively incompatible graphs can be very contrast in terms of their hubs and authority vertices which can lead to a information dilution and loss of quality. In such cases, a wise decision is to only choose the most effective – the most compatible – set of homogeneous networks to merge from multiple homogeneous graphs; this decision lies with the data scientist. To effectively measure the compatibility of graphs, we use the hub and authority matrices from both graphs. Close observation on HITS [25] algorithm reveals that it is an iterative power method to compute the dominant eigenvector for $M \cdot M^T$ and for $M^T \cdot M$ where matrix M is an adjacency matrix of a graph. Hub matrix is $\mathcal{H} = M \cdot M^T$ and authority matrix is $\mathcal{A} = M^T \cdot M$. Also, constant initialization of hub/authority scores enables us to perform power iteration on $\mathcal{H}$ and $\mathcal{A}$ and choose matrices from any iteration. Let $\mathcal{H}_r^p, \mathcal{H}_r^w$ and $\mathcal{A}_r^p, \mathcal{A}_r^w$ be the hub and authority score matrices of two homogeneous graphs on ROI R.

Finding similarity or distance with labeled graphs is an easy task and we can leverage simple methods like edit distances, matrix similarity or even complex methods like coupled vertex-edge scoring [53], MCES [37] etc. For our model, we use Frobenius distance between two matrices and they qualify for merge if the sum of distance is less than some positive value ϕ.

$$F_{A,B} = \|A - B\|_F = \sqrt{\sum_{i,j} (A_{i,j} - B_{i,j})^2} \tag{6}$$

$$F_{\mathcal{H}_r^p, \mathcal{H}_r^w} + F_{\mathcal{A}_r^p, \mathcal{A}_r^w} \leq \phi \tag{7}$$

Similar to G_r' homogeneous graph of ROIs, we construct G_p' and G_w' by merging (G_p^r, G_p^w) and (G_w^r, G_w^p, G_w), respectively.

3.2.3 Community-aware Random Walks

Homogeneous graphs constructed from bipartite networks is used to generate a corpus of several random graphs. DeepWalk [35] generates such random walk and utilize it for learning embedding. BiNE [16] addresses issues that DeepWalk [35] does not capture the characteristic of the real world network because the distribution of vertices in random walks, and the graph network does not match. One solution is to generate random walks based on the importance of vertices measured with hubs and authority score of vertices.

Community is defined as a subset of vertices within the graph such that connections between the vertices are denser than connections with the rest of the network [36]. If the number of connections or reachability between vertices within a very few hops is high, then they must have a stronger bond. In a real-world scenario, we often have edges that act as bridges between communities or sub-communities. Often sparsity and lack of information in training data are responsible for the appearance of bridges within a community. Even if there is a moderate number of bridges, centrality biased random walks will seldom connect them. We propose a δ-hop community-aware random walk where a step in the random walk can

Algorithm 1: Community-aware Random Walk

```
Input: Graph $G'_u$, Sequence Params $(\mu, \sigma)$, maxStart, Walk mutation probability $\alpha$ ;
Output: $\mathcal{D}_u$
Adjacency matrix $M$ = Adjacency($G'_u$) ;
$H$ = Centrality($M$);
3-hop Adjacency Matrix $M^3$;
foreach vertex $u_i \in U$ do
    numStart = max($H(u_i)\times$ maxStart, 1);
    for i=1 to numStart do
        # Choose length $l$ with Normal Distribution;
        Length $l = N(\mu, \sigma)$;
        Sequence $S = \{u_i\}$;
        cur = $u_i$ ;
        for $k = 2$ to $l$ do
            rand = RandomFloat((0,1]);
            if  $0 <$ rand $\leq \alpha$ then
                $u_k$ = MutateRandomWalk($M^3$, cur) ;
            end
            else
                $u_k$ = RandomWalk($M$, cur) ;
            end
            cur = $u_k$;
            Add cur to Sequence $S$ ;
        end
        Add Sequence $S$ to $\mathcal{D}_u$
    end
end
return $\mathcal{D}_u$;
```

mutate to a jump with probability α within δ-hop connected community.

The motivation of a δ-hop community is to include strongly/well-connected bridges and avoid weak connected community bridges. We used M^3, 3-hop is the least number of hops such that an internal node from a well-connected community can reach an internal node of another community via a bridge, where M is a adjacency matrix.

Hence, it is straightforward to follow that with a low $\delta = 3$ and a low step-jump mutation probability $\alpha = 0.1$ the jump likely remains within the community but alleviates the moderately connected community problem. Like other biased random walk model following "rich gets richer" principle, our mutated step-jump act as a welfare strategy in the algorithm.

Algorithm 1 presents the summarized community-aware random walk to prepare corpus $\mathcal{D}_u$ from graph homogeneous G'_u. Statistic suggests that mean length of sentences in English varies between 20-25 words and follows normal distribution [52]. Technical writing sentences are typically shorter. We take the inspiration from it and use normal distribution with mean $\mu = 15$ and standard deviation $\sigma = 10$ to generate length of sequences in corpus $\mathcal{D}_u$. Starting a sequence with a vertex depends on its popularity (centrality) but we also limit it to a maximum of 5 with variable *maxStart*.

3.2.4 Corpus Generation

Following the community-aware random walk on G'_r, G'_p, G'_w we obtain corpuses $\mathcal{D}_r, \mathcal{D}_p, \mathcal{D}_w$ respectively by using Algorithm 1.

For a sequence $\mathcal{S}$ in corpus $\mathcal{D}_r$ an ROI r_i positioned at index c in $\mathcal{S}$ is represented as r_i^c. In a sequence $\mathcal{S}$ a context of m from c will be the ROIs positioned from $c-m$ to $c+m$ i.e., $\{r_\odot^{c-m}, r_\odot^{c-m+1}, \ldots, r_\odot^{c}, r_\odot^{c+1}, \ldots, r_\odot^{c+m}\}$ where $\odot$ is in range $[1, |P|]$. We can now apply the skip-gram model on corpuses similar to the technique used in Word2Vec [29] embedding to optimize each embedding entity. To optimize the embedding for ROIs $\vec{\boldsymbol{r}}$, POIs $\vec{\boldsymbol{p}}$, and Words $\vec{\boldsymbol{w}}$ we should minimize the expressions for objective functions O'_r, O'_p, and O'_w respectively. It is to note that for each entity, as we create an embedding vector, we also need to assign a corresponding *context vector* for that entity.

$$
\begin{gathered}
\text{minimize } O'_r = - \sum_{r_i^c \in \mathcal{S} \wedge \mathcal{S} \in \mathcal{D}_r} \sum_{j=0, j\neq m}^{2m} \log \overline{\Pr}(r_\odot^{c-m+j} | r_i^c) \\
\overline{\Pr}(r_\odot^{c-m+j} | r_i^c) = \frac{\exp\left(\overrightarrow{\boldsymbol{r_i^c}}^T \cdot \overrightarrow{\boldsymbol{\varphi_\odot^{c-m+j}}}\right)}{\sum_{k=1}^{|R|} \exp\left(\overrightarrow{\boldsymbol{r_i^c}}^T \cdot \overrightarrow{\boldsymbol{\varphi_\odot^{k}}}\right)} \\
\text{where } \overrightarrow{\boldsymbol{\varphi_\odot^{c}}} \text{ is the context vector for } r_\odot^c.
\end{gathered}
\tag{8}
$$

Similarly, we optimize for POIs $\vec{\boldsymbol{p}}$ with function O'_p.

$$\begin{aligned}
&\text{minimize } O'_p = -\sum_{p_i^c \in \mathcal{S} \wedge \mathcal{S} \in \mathcal{D}_p} \sum_{j=0, j\neq m}^{2m} \log \overline{\Pr}(p_\odot^{c-m+j} | p_i^c) \\
&\overline{\Pr}(p_\odot^{c-m+j}|p_i^c) = \frac{\exp(\overrightarrow{\boldsymbol{p_i^c}}^T \cdot \overrightarrow{\boldsymbol{\varrho_\odot^{c-m+j}}})}{\sum_{k=1}^{|P|} \exp(\overrightarrow{\boldsymbol{p_i^c}}^T \cdot \overrightarrow{\boldsymbol{\varrho_\odot^{k}}})} \\
&\qquad \text{where } \overrightarrow{\boldsymbol{\varrho_\odot^{c}}} \text{ as the context vector for } p_\odot^c.
\end{aligned} \tag{9}$$

Finally, we optimize for Words $\vec{\boldsymbol{w}}$ with function O'_w.

$$\begin{aligned}
&\text{minimize } O'_w = -\sum_{w_i^c \in \mathcal{S} \wedge \mathcal{S} \in \mathcal{D}_w} \sum_{j=0, j\neq m}^{2m} \log \overline{\Pr}(w_\odot^{c-m+j} | w_i^c) \\
&\overline{\Pr}(w_\odot^{c-m+j}|w_i^c) = \frac{\exp(\overrightarrow{\boldsymbol{w_i^c}}^T \cdot \overrightarrow{\boldsymbol{\vartheta_\odot^{c-m+j}}})}{\sum_{k=1}^{|W|} \exp(\overrightarrow{\boldsymbol{w_i^c}}^T \cdot \overrightarrow{\boldsymbol{\vartheta_\odot^{k}}})} \\
&\qquad \text{where } \overrightarrow{\boldsymbol{\vartheta_\odot^{c}}} \text{ as the context vector for } w_\odot^c.
\end{aligned} \tag{10}$$

3.3 Negative Sampling

The conditional probability $\overline{\Pr}(v_j|u_i)$ from Equation 3,4 and $\overline{\Pr}(u_j|u_i)$ from Equation 8,9,10 is computationally expensive since it would need to sum over the entire set of vertices. The state-of-the-art method to empirically estimate them is via *negative sampling* (e.g., as in specified in [29]), where the denominator is estimated by sampling random vertices. The numerator (defined by explicitly similar vertices) can be calculated directly.

In particular, negative sampling helps to learn a better embedding by selecting negative vertices that have significant probability difference, yet are closely connected vertices. Our negative sampling method uses popularity biased method which helps in learning faster but also alleviates gradient vanishing issues [6]. We use the concept of transition probabilities in random walk from one vertex to another and this strategy perfectly replicates the popularity/ranking based system which we leverage for negative sampling [55]. In a random walk starting from vertex u_i adjacent to vertex u_j the probability of reaching from u_i to u_j is defined as the ratio of the weight of the edge (u_i, u_j) over the sum of weights on all adjacency edges of vertex u_i. We compute the (i, j)th cell of transition matrix T from the adjacency matrix M of graph as

$$T(u_i, u_j) = \frac{M_{u_i,u_j}}{\sum_{k=1}^{|M_{u_i}|} M_{u_i,u_k}} \tag{11}$$

where M_{u_i,u_j} is weight of edge between u_i and u_j.

Naturally T is a right stochastic matrix. We also make sure that self-loops, if they initially exists, are removed from the matrix. Based on the matrix T we perform a δ-hop random walk by power iteration T^δ. For some dense graphs the matrix can converge and reach a steady state distribution in few hops. For our purpose we restrict the δ to $\delta_{\max} = 5$. The row u_i of $T^{\delta_{\max}}_{u_i}$ act as a noise distribution matrix for selection of negative candidates for target vertex u_i. We define the K negative samples for target u_i as $N^K_{G_u}(u_i)$.

Following the negative sampling technique for homogeneous graphs, we need to extend this technique for incorporating bipartite graphs as well. Firstly, we assume the prevalence of transitive property for bipartite graphs to model hops between same type of vertices i.e. if u_i is connected to v_k and then v_k is connected to u_j then we assume existence of edge between u_i and u_j in graph G_{uv}. The weights of edge i.e. $\varpi(u_i, u_j) = \sum_{v_k \in V} \varpi(u_i, v_k) \cdot \varpi(v_k, u_j)$. After we have defined the edges and weights between connected u_is and u_js, it is easy to obtain T. Thereafter, $\delta_{\max}$-hop and $T^{\delta_{\max}}$ noise matrix is obtained to perform negative sampling on same seed and target type vertices, we dub this as *homogeneous negative sampling*. For a seed vertex u_i in bipartite graph K negative samples $N^K_{G_{uv}}(u_i)$ is obtained from noise distribution row $S(u_i)$ in S where

$$S(u_i) = \{T^{\delta_{\max}}(u_i, u_j) : \forall u_j \in U\} \tag{12}$$

For negative sampling on bipartite graphs where the seed vertex is different from target sample vertices, which we dub as *heterogeneous negative sampling*, e.g. seed u_i to target v_l, we apply the usual transition probabilities on the already obtained noise matrix $T^{\delta_{\max}}$. When a vertex u_i connected to u_j in $\delta_{\max}$-hop, all the adjacent vertices of u_j say $V' = \{v_l | e_{u_j,v_l} \in E_{uv})\}$ are now considered for heterogeneous negative sampling. The entry of (u_i, v_l) cell in noise distribution matrix of bipartite graphs is calculated as

$$S(u_i, v_l) = \sum_{u_j \in U} T^{\delta_{\max}}(u_i, u_j) \cdot \frac{M_{u_j,v_l}}{\sum_{v_m \in V} M_{u_j,v_m}} \tag{13}$$

where $\frac{M_{u_j,v_l}}{\sum_{v_m \in V} M_{u_j,v_m}}$ is the transition probability from u_j to v_l.

```
Algorithm 2: Embedding Training
Input: Graph $\mathcal{G}$, Training Samples N, Random Walk Params ($\mu, \sigma$,$maxStart$,$\alpha$), negative sample $K$ ;
Output: ROI embedding $\vec{\boldsymbol{r}}$,POI embedding $\vec{\boldsymbol{p}}$, Word embedding $\vec{\boldsymbol{w}}$ ;
Create $G_{rp}, G_{rw}, G_{pw}, G'_r, G'_p, G'_w$ from $\mathcal{G}$;
$\mathcal{D}_r$ = Community-aware Random Walk($G'_r$,$\mu, \sigma$,$maxStart$,$\alpha$);
$\mathcal{D}_p$ = Community-aware Random Walk($G'_p$,$\mu, \sigma$,$maxStart$,$\alpha$);
$\mathcal{D}_w$ = Community-aware Random Walk($G'_w$,$\mu, \sigma$,$maxStart$,$\alpha$);
Perform negative sampling $N_{G_{rp}}$ from $\mathcal{D}_r$, $\mathcal{D}_p$ and $G_{rp}$.; $N_{G_{rw}}$ from $\mathcal{D}_r$, $\mathcal{D}_w$ and $G_{rw}$.; and $N_{G_{pw}}$ from $\mathcal{D}_p$, $\mathcal{D}_w$ and $G_{pw}$.;
Perform negative sampling $N_{G_r}$ from $\mathcal{D}_r$ and $G_r$.; $N_{G_p}$ from $\mathcal{D}_p$ and $G_p$.; and $N_{G_w}$ from $\mathcal{D}_w$ and $G_w$.;
Initialize embedding vectors ROIs $\vec{\boldsymbol{r_i}}$, POIs $\vec{\boldsymbol{p_j}}$, Words $\vec{\boldsymbol{w_k}}$ ; and context vectors ROIs $\vec{\boldsymbol{\varphi_i}}$, POIs $\vec{\boldsymbol{\varrho_j}}$, Words $\vec{\boldsymbol{\vartheta_k}}$ ;
while iteration = 1 to N do
    Sample edge $(r_i, p_j)$ from $G_{rp}$, pick $N^K_{G_{rp}}(r_i)$, $N^K_{G_{rp}}(p_j)$;
        Update ROI $\vec{\boldsymbol{r_i}}$ and context POI $\vec{\boldsymbol{\varrho_n}}, p_n \in N^K_{G_{rp}}(r_i)$ & update POI $\vec{\boldsymbol{p_j}}$ and context ROI $\vec{\boldsymbol{\varphi_n}}, r_n \in N^K_{G_{rp}}(p_j)$ ;
    Sample edge $(r_i, w_j)$ from $G_{rw}$, pick $N^K_{G_{rw}}(r_i)$, $N^K_{G_{rw}}(w_j)$;
        Update ROI $\vec{\boldsymbol{r_i}}$ and context Word $\vec{\boldsymbol{\vartheta_n}}, w_n \in N^K_{G_{rw}}(r_i)$ & update Word $\vec{\boldsymbol{w_j}}$ and context ROI $\vec{\boldsymbol{\varphi_n}}, r_n \in N^K_{G_{rw}}(w_j)$ ;
    Sample edge $(p_i, w_j)$ from $G_{pw}$, pick $N^K_{G_{pw}}(p_i)$, $N^K_{G_{pw}}(w_j)$;
        Update POI $\vec{\boldsymbol{p_i}}$ and context Word $\vec{\boldsymbol{\vartheta_n}}, w_n \in N^K_{G_{pw}}(p_i)$ & update Word $\vec{\boldsymbol{w_j}}$ and context POI $\vec{\boldsymbol{\varrho_n}}, p_n \in N^K_{G_{pw}}(w_j)$ ;
    Sample edge $(r_i, r_j)$ from $G_r$, pick $N^K_{G_r}(r_i)$;
        Update ROI $\vec{\boldsymbol{r_i}}$ and context $\vec{\boldsymbol{\varphi_n}}, r_n \in N^K_{G_r}(r_i)$ ;
    Sample edge $(p_i, p_j)$ from $G_p$, pick $N^K_{G_p}(p_i)$;
        Update ROI $\vec{\boldsymbol{p_i}}$ and context $\vec{\boldsymbol{\varrho_n}}, p_n \in N^K_{G_p}(p_i)$ ;
    Sample edge $(w_i, w_j)$ from $G_w$, pick $N^K_{G_w}(w_i)$;
        Update Word $\vec{\boldsymbol{w_i}}$ and context $\vec{\boldsymbol{\vartheta_n}}, w_n \in N^K_{G_w}(w_i)$ ;
end
return ROI embedding $\vec{\boldsymbol{r}}$, POI embedding $\vec{\boldsymbol{p}}$, Word embedding $\vec{\boldsymbol{w}}$;
```

For each edge (u_i, v_j) in a graph with target vertex u_i and K negative samples we follow the conditional probability approximation $\overline{\Pr}(v_j|u_i)$, where $\vec{\boldsymbol{\varsigma_j}}$ is the context vector for v_j as follows.

$$\overline{\Pr}(v_j|u_i) = \log\sigma(\vec{\boldsymbol{\varsigma_j}} \cdot \vec{\boldsymbol{u_i}}) + \sum_{v_n \in N^K_{G_{uv}}(u_i)} \log\sigma(-\vec{\boldsymbol{\varsigma_n}} \cdot \vec{\boldsymbol{u_i}}) \tag{14}$$

Similarly, for $\overline{\Pr}(u_j|u_i)$, where $\vec{\boldsymbol{\varkappa_j}}$ is the context vector for u_j as follows.

$$\overline{\Pr}(u_j|u_i) = \log\sigma(\vec{\boldsymbol{\varkappa_j}} \cdot \vec{\boldsymbol{u_i}}) + \sum_{u_n \in N^K_{G'_u}(u_i)} \log\sigma(-\vec{\boldsymbol{\varkappa_n}} \cdot \vec{\boldsymbol{u_i}}) \tag{15}$$

3.4 Optimization and Model Update

The intuitive solution for optimization is to minimize the sum of all objective functions. A more complex solution for multiobjective optimization can be applied. However, choosing a multiobjective optimization in embedding scenario requires more studies and can be presented as a separate research work on its own. Having said that we use non-weighted linear combination of each optimization expresions from equations 4, 8, 9, and 10 to make a single global optimization.

$$O = O_{rp} + O_{rw} + O_{pw} + O'_r + O'_p + O'_w \tag{16}$$

We present our tripartite joint optimization in Algorithm 2. In the preparation phase, community-aware random walks generate corpora $\mathcal{D}_r, \mathcal{D}_p, \mathcal{D}_w$, negative sampling module prepares noise distribution matrices. In the joint embedding training phase, edges are sampled from each graph simultaneously, and updates embedding vectors along with the context vectors using the Stochastic Gradient Descent algorithm.

The complexity of the training depends on the density/sparsity of the graph network. To avoid expensive computation of centrality and δ-hop adjacency matrix, we perform walks on the graph based on degree centrality. The context size for a vertex is $b \cdot m$; where b is the batch size much less than the maximum degree of the vertex, and m is context defined in section 3.2.4. Overall the computation complexity of our algorithm is $O(|E_{rp} + E_{rw} + E_{pw}| \cdot b \cdot m \cdot (ns + 1))$, where ns is the number of negative samples.

TNE supports increment updates as we collect new datasets from social networks and create a new information graph or update the old information graph. In this case, the embeddings previously generated from TNE should be used instead of random initialization of embedding vectors. Hyperparameter tuning, such as the learning rate, should be tweaked based on the age of the previously trained dataset and volume of the new dataset. With the increasing volume of new data and the aging of the previous dataset, the learning rate can gradually increase for optimal performance.

4 Experiments

In this section, we first describe our real-world dataset based on New York City (NYC) used in our experiments. We then present five experiments we performed exhibiting multi-faceted effectiveness of ROI embedding with TNE on spatial correlation, semantic association, and predictive capabilities. A summary of the experiments are as follows:

1. *Model validation with POI classification:* We perform this POI embedding experiment to validate the fact that TNE, TNE_NW, and state-of-the-art baselines can perform this task equally well as expected. In the follow-up experiments, we show that baselines methods cannot perform at par with TNE or TNE_NW on ROI experiments validating the necessity for TNE.
2. *Geospatial affinity of ROIs:* This experiment evaluates all the models on ROI embeddings, whether it can preserve the spatial correlations among ROIs in the embedding space as the original data. By spatial correlation, we mean neighboring or spatially overlapping ROIs.
3. *Semantic category annotation of ROIs:* In this experiment, we perform a ranking evaluation task with category annotation from ROI embeddings and crowdsourced ground truth results. We use Normalized Discounted Cumulative Gain (NDCG) [44] metric as the measure of performance.
4. *Semantic category difference from ROI embedding:* This experiment is similar to the previous experiment with a distinction here that we try to evaluate the semantic difference between a pair of ROIs from their embedding.
5. *Popularity Prediction of Regions:* We introduce Region Popularity Prediction experiment with the simplest of regression models to demonstrate that ROI embedding with TNE_NW, TNE can capture features better than extended baselines along with temporal features. The aim is not to overcomplicate experiment with complex models aiming lowest error but to show perceptible differences even with simple feature-based models.

4.1 Dataset

The dataset imitates the information graph $\mathcal{G}$, we presented in Figure 2. Also, as described in Section 2 our real world dataset consist of three entities (a) POI (b) ROI (c) Word. We will release the anonymized processed version of dataset adhering to the copyright of the sources for the growth of research work in this field.

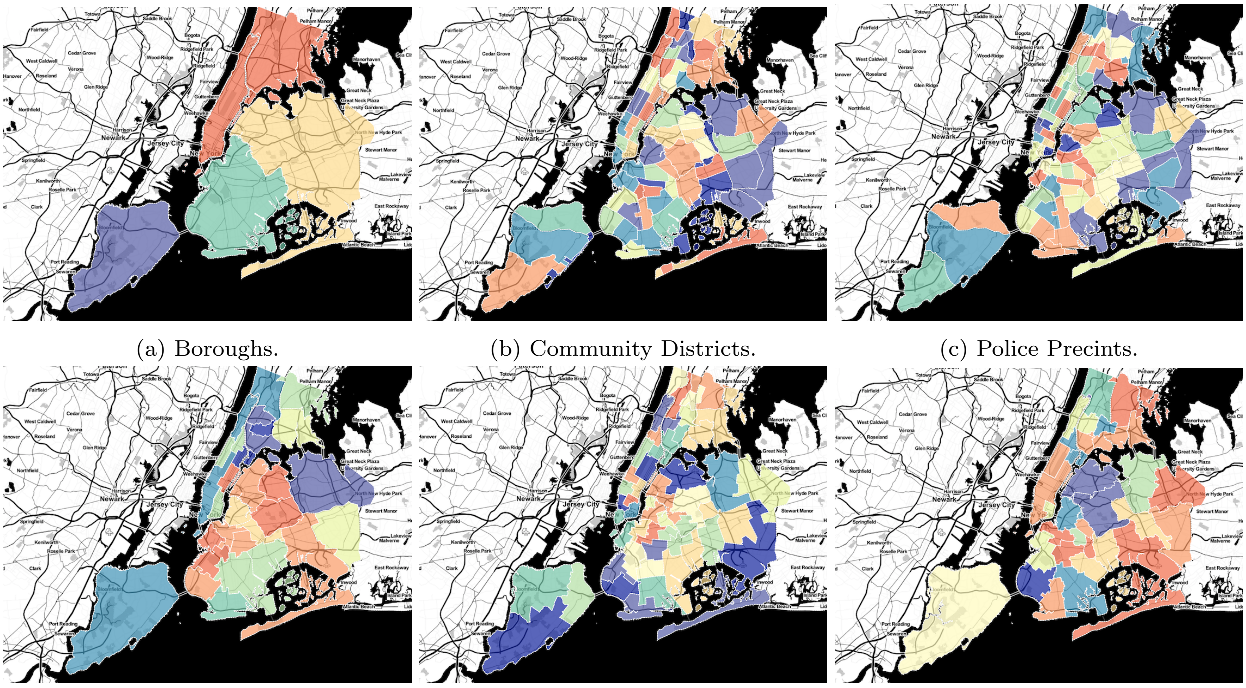
POI-Word Data. We used the check-in dataset from [49] and NYC government site [32] to collect POI dataset. Our dataset comprise of 38,008 POIs. Each POI is associated with geolocation, name, category, description, and comments. The words from name, description, and all available comments from each POI is cleaned and tokenized in a preprocessing step. The association of words with POIs are used to create edges between them. As mentioned in Section 2.1, the weights of the POI-Word edges is calculated based on their TF-IDF score.

ROI-POI Data. The ROI data is obtained from the publicly available GeoJSON [31] of NYC that are demarcated with multi-polygonal features. Each GeoJSON has a unique geographical division of NYC based on its type. Figure 3 shows some geographical divisions of NYC, such as boroughs, city councils, election districts, fire battalions, police precints districts, health districts, etc. All geographical divisions consist of several non-overlapping ROIs, and each of them is treated as a separate and unique ROI in our dataset. Overall we have 12 different geographical divisions/districts as stated in Table 3 along with the number of ROIs from that division. The total number of ROIs in our dataset is 456.

A POI is associated with an ROI iff the geolocation of the POI is within the polygonal boundary of the multi-polygonal spatial feature. It is notable that for a non-overlapping set of ROIs; POIs

Table 3: Geographical Divisions of NYC.

ID	Division Type	ROIs
01	Boroughs	5
02	City Council	51
03	Community Districts	71
04	Congressional Districts	13
05	Fire Battalions	49
06	Fire Divisions	9
07	Health Centers Districts	30
08	Municipal Courts Districts	28
09	Police Precints	76
10	School Districts	33
11	State Assembly	65
12	State Senate	26



(a) Boroughs. (b) Community Districts. (c) Police Precints.

(d) Health Centers Divisions. (e) Fire Battalions Divisions. (f) School Districts.

Fig. 3: ROI dataset sample: Different Geographical divisions of NYC.

1 month			6 months		
ROI	*Word*	*Score*	*ROI*	*Word*	*Score*
12	empirestatebuilding	0.019	**07**	**empirestatebldg**	**0.030**
07	**empirestatebuilding**	**0.013**	**07**	**empirestatebuilding**	**0.025**
00	empirestatebuilding	0.011	12	empirestatebuilding	0.012
06	empirestatebuilding	0.010	06	empirestatebuilding	0.011
07	empirestatebldg	0.008	08	empirestatebuilding	0.007
02	empirestatebuilding	0.007	02	empirestatebuilding	0.005
71	empirestatebldg	0.006	06	empirestatebuilding	0.003
01	**brooklynbridge**	**0.081**	**53**	**brooklynbridge**	**0.079**
53	**brooklynbridge**	**0.046**	**53**	**brooklynbridgepark**	**0.091**
53	**brooklynbridgepark**	**0.041**	**01**	**brooklynbridge**	**0.074**
39	brooklynbridge	0.035	00	brooklynbridge	0.019
12	brooklynbridge	0.031	03	brooklynbridge	0.017
08	brooklynbridge	0.029	04	brooklynbridge	0.008

Table 4: Sample comparison of ROI & TF-IDF score for 1 month and 6 month data with ROI Map.

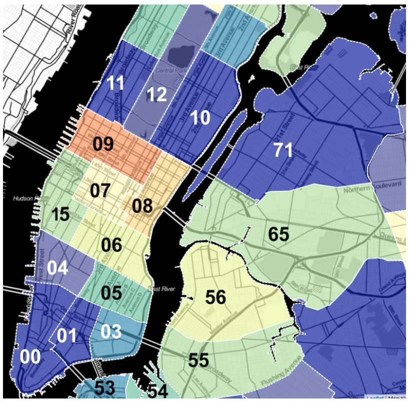


Fig. 4: ROI from Police Precints Districts of NYC with IDs.

will create a many-one onto relation function with ROIs. However, introducing overlapping ROIs makes the information graph $\mathcal{G}$ interesting because shared POIs among two or more overlapping ROIs increases the complexity of the graph. The associated weight of the edges in ROI-POI graph is assigned a value of 1.0.

ROI-Word Data. The relationship between ROI and Word is obtained from the geotagged tweets collected over a period of time. Similar to the technique used with POI-Word pair, the weight of edge between an ROI-Word is determined from the TF-IDF score. First, we used 1% sample tweet stream from twitter to collect our geotagged documents for one month and prepare a corpus of documents (each document associated with an ROI). On analyzing the twitter stream and performing TF-IDF on the corpus, it revealed that one month of 1% sample stream is not enough to extract meaningful TF-IDF scores and information out of blabber, chores, and chatter of tweets. To alleviate the problem we used 6 months of 1% sample twitter stream. It is still a very feasible approach as 6 months of 1% sample roughly equates to 2-3 weeks of original twitter stream or firehose API of Twitter.

We present examples of the TF-IDF scores w.r.t. ROIs from Police Precincts division for 1 month and 6 months of data in Table 4 for two famous attractions of NYC i.e. *Empire State Building* and *Brooklyn Bridge*. For 1 month of data and word `empirestatebuilding`, ROI 12 have maximum TF-IDF score of 0.019. However, the location of *Empire State Building* suggests ROI 07 should have the maximum score, which ranks second in Table 4. It is also notable that TF-IDF scores for 1 month does not suggest good spatial correlation. On contrast the result with 6 months of data shows significant correlation of TF-IDF score and true ROI location for Empire State Building i.e. ROI 07. Furthermore, the TF-IDF scores for the same are in accordance with the neighborhood ROIs showing strong geospatial correlation.

Another interesting trend can be seen with *Brooklyn Bridge*, where the true spatial location is ROI 01 and 53. For 1 month of data though the top TF-IDF scores are in accordance with ground truth ROI location but the scores are very close to the ROIs that are not near to *Brooklyn Bridge* (i.e. ROI 53: 0.041; ROI 39: 0.035). Whereas, a clear disparate between TF-IDF scores of ground truth ROIs (53 & 01) and other ROIs (00, 03 & 04) with 6 months of data. These examples explains and supports our decision of using 6 months of geotagged tweets.

Region Popularity Data. Region popularity data is collected from the New York check-in dataset [49], which contains 227,428 check-ins from Foursquare for period of April 2012 to February 2013. We score the popularity of a region from the number of check-ins.

4.2 TNE **validation with POI Classification**

This experiment evaluates POI embedding from our model and baselines. The aim of this experiment is to validate that our model is consistent in learning POI embedding as other state-of-the-art work. In this experiment we expect all the methods to perform equally well.

Our POI dataset have a ground truth category for each POI which has been collected from the data source. It is worth mentioning that Table 2 presents all nine top level categories for our POI dataset.

First, we present *(i) k-Nearest Neighbor Classification* to evaluate POI embedding of all the models. Then we use *(ii) t-SNE visualization* to notice the macro- and micro-structure of embeddings.

To boost our learning process, we initialized word embeddings with pre-trained GloVe [34] embedding. We used Glove vectors of words from *description* of POIs for POI embedding initialization. However, all the ROI embeddings are always initialized with random vectors. Our justification for initialization is to utilize full resources and information available in hand, rather than spending more iterations on learning from random initialization.

4.2.1 k-Nearest Neighbor Classification

We trained our *k-Nearest Neighbor (k-NN) Classifier* on 70% of embedding and evaluated on the rest of the embedding data. The dimension of embedding was kept 100 and k stands for the number of nearest neighbors considered for k-NN classification. From the result presented in Table 5, we see that GE_poi, TNE_nw, and TNE performed similarly with 96% accuracy in determining top category, whereas, BiNE achieves more than 95% for k-NN with $k \geq 3$. It verifies that TNE achieves comparable state-of-the-art performance with GE_poi. We have not included CrossMap result in Table 5 because CrossMap does not produce POI embedding.

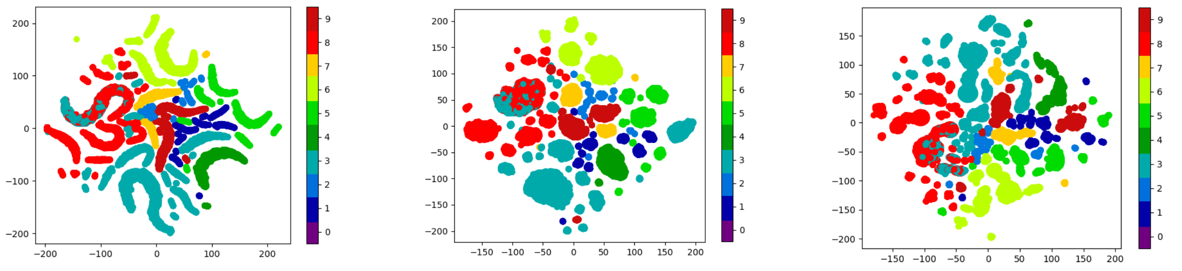
4.2.2 t-SNE visualization

To reveal subtlety of the POI embedding and explore macro- and micro-structure, we perform t-SNE on the high dimensional POI embedding. We color each POI in accordance to the top category mentioned in Table 2. Figure 5 shows how the POI embedding changes from training iteration 10 and 40 for TNE. Figure 5a shows different category points are much nearer and somewhere overlaps with one another. The scenario of such overlaps and distance between dissimilar category cluster improves with more iteration in Figure 5b. We also present t-SNE of GE_poi in Figure 5c.

Though our k-NN classification and t-SNE yield good performance for top level category or macro-structure, our experiment did not feature so well with subcategories. In our microscopic evaluation of the t-SNE embedding by analyzing the embedding of subcategories in Figure 6. For this experiment we have taken all the POIs with top category as *Travel & Transport* and performed t-SNE on it. The colors of

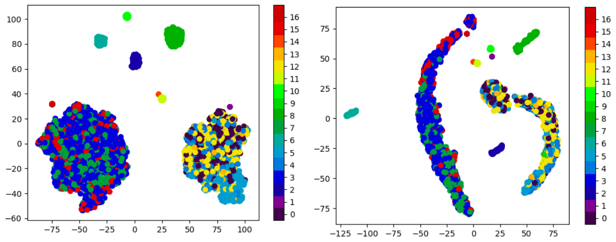
Table 5: POI k-NN Classification Accuracy

Model \ k	1	3	5	10
GE_poi	0.963	0.966	0.966	0.968
BiNE	0.926	0.954	0.957	0.955
TNE_wcr	0.923	0.943	0.952	0.951
TNE_nw	0.959	0.955	0.967	0.962
TNE	0.964	0.964	0.965	0.967



(a) TNE Iteration 10. (b) TNE Iteration 40. (c) GE_POI Iteration 40.

Fig. 5: t-SNE visualization of POI embedding.



(a) TNE. (b) GE_POI.

Fig. 6: t-SNE visualization of Travel & Transport.

Table 6: A category similarity analysis for a similar ROI pair, ROI 11000043 from Staten Island and ROI 05000045 from Brooklyn, New York City

C ROI	Arts & Entertainment	Outdoors Activities	Shops & Services	Nighlife Spot
11000043	0.3445	0.2992	0.0971	0.1583
05000045	0.3481	0.2542	0.1032	0.1855

the POIs in Figure 6 is based on the subcategories. Here we provide the list of the subcategories for *Travel & Transport* and order them with the color number in the t-SNE visualization. *0. Airport 1. Bike Rental / Bike Share 2. Boat or Ferry 3. Bus Station 4. General Travel 5. Hotel 6. Light Rail Station 7. Metro Station 8. Moving Target 9. Pier 10. Rental Car Location 11. Rest Area 12. Road 13. Taxi 14. Tourist Information Center 15. Train Station 16. Travel Lounge.*

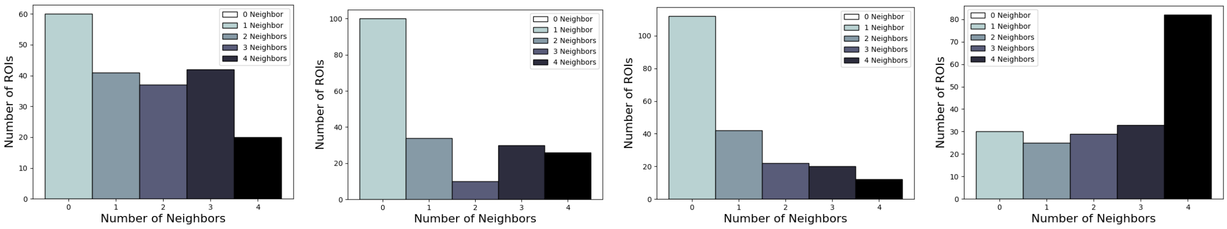
It is clear from Figure 6, that the POI embedding of subcategories are overlapping for both GE_POI and TNE. The close association among POIs under same top level category might explain such embedding phenomenon in semantic space. However, it might be worth to look into features of such intra-categorical POIs in future work.

4.3 Geospatial Affinity of ROIs

In this section, we evaluate the ROI embedding based on the geospatial affinity among ROIs. The intended scenario is to obtain similar embeddings for ROIs having geospatial affinity i.e. (a) overlapping region, and (b) neighboring region.

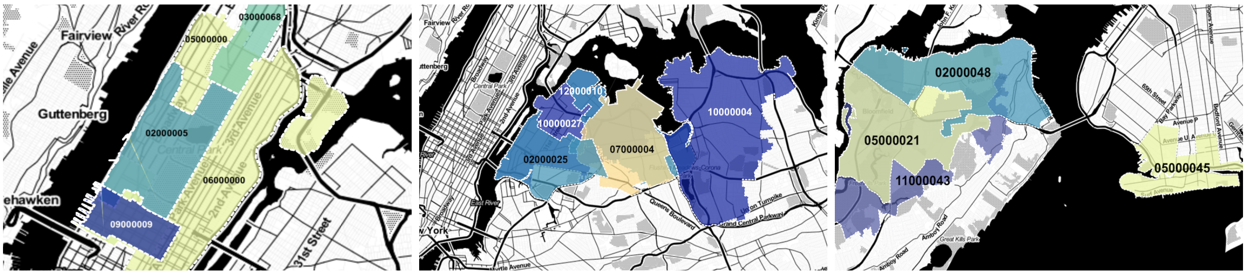
We randomly selected 200 ROIs and analyzed 4 nearest neighbors of each ROI from our embedding with crowdsourced ground truth. Human judgment is used to find out whether the nearest neighbors ROIs predicted from embedding have any geospatial affinity or not with the queried ROI. We build a website with geographical map for crowdsourcing and to facilitate this process. Ideally we would want more ROIs with 3 to 4 geospatially overlapped neighbors from the k-NN result from embedding space with $k = 4$. It is worth to mention that our dataset has 12 different geographical division, that means each ROI have many (atleast 10) geospatially overlapped ROIs. In plot of Figure 7d, we show the number of ROI neighbors that have geospatial affinity for TNE. The last histogram bar with black color shows that out of 200 ROIs more than 80 ROIs have 4 neighbors with geographical overlapping region or neighboring boundary for TNE. We performed similar analysis on GE_POI the number of ROIs with 4-NN is comparatively low (only 10%) as shown in Figure 7a, compared to 40% with our model in Figure 7d. The results for CrossMap and BiNE are far worst with almost 50% and 55% of the ROIs with zero geospatially overlapped neighbors respectively as shown in Figure 7b and 7c. From this result we can strongly deduce that our embedding preserves geospatial affinity in its embedding which other baseline approaches cannot.

Figure 8 provides three examples of the geospatial affinity (with query ROIs 06000000, 07000004, and 11000043) obtained by utilizing nearest neighbor technique on ROI embedding. In Figure 8a shows nearest neighbors of ROI 06000000 from embedding (05000000, 03000068, 09000009, 02000005). Similarly, Figure 8b shows nearest neighbors of ROI 07000004 as (10000027, 12000010, 02000025, 10000004). The interesting observation on the Figure 8c for nearest neighbors for ROI 11000043 in Staten Island is that it finds a ROI 05000045 located in Brooklyn to be similar. More detailed observation on both the ROIs reveals that they are similarly popular with Arts and



(a) GE_POI. (b) CrossMap. (c) BiNE. (d) TNE.

Fig. 7: Geospatial Affinity from ROI Embedding, more the number of geospatial neighbors in k-NN the better.



(a) 4-NN of ROI 06000000 in Manhattan, New York City. (b) 4-NN of ROI 07000004 in Queens, New York City. (c) 3-NN of ROI 11000043 in Staten Island, New York City.

Fig. 8: Examples of Geospatial Affinity of ROIs with ROI Embedding and Nearest Neighbors.

Entertainment POIs, Outdoor activities as obtained from the cosine similarities of the embeddings. Table 6 presents the similarity scores of the above mentioned ROIs for some semantic categories. We will discuss more on the technical methods on obtaining it in Section 4.4.

4.4 Semantic Category Annotation of ROIs

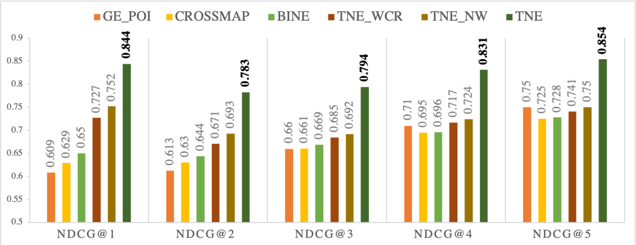
In this section, we present the analysis of ROI embedding on *semantic category annotation*. First, we show an example of semantic annotation in Table 7 for ROI 09000056. The geospatial location of ROI 09000056 *Greenpoint, Brooklyn, NYC* is presented in the map along with Table 4 as ROI 56. The rank of categories in Table 4 suggests *Greenpoint* has considerable shops & services locations, recreation parks and residential complexes. To verify our prediction, we tallied the rank with human raters who used Foursquare [14], NYC government site [9] and ArcGIS [2] maps, Twitter [42] and Wikipedia [46] for ground truth information. Crowdsourced ground-truth semantic categories of ROIs are ranked into three levels (1) *low relevant level* (2) *moderately relevant level,* (3) *highly relevant level.* Crowdsourced information for ROI 09000056 suggests that there are many good shops, McCarren Park for outdoor activities and residential complexes. This information aligns with top 3 categories of semantic category annotation (a) Shops & Services (b) Outdoors & Recreation (c) Residential.

For a comprehensive analysis, we crowdsourced ground-truth categories with human raters for 100 random ROIs with category levels 3,2, and 1. We compare ground-truth against the *semantic category annotations* obtained from the embedding. We converted it into a ranking problem. In an ideal case,

Table 7: ROI 09000056: Semantic Category Annotation

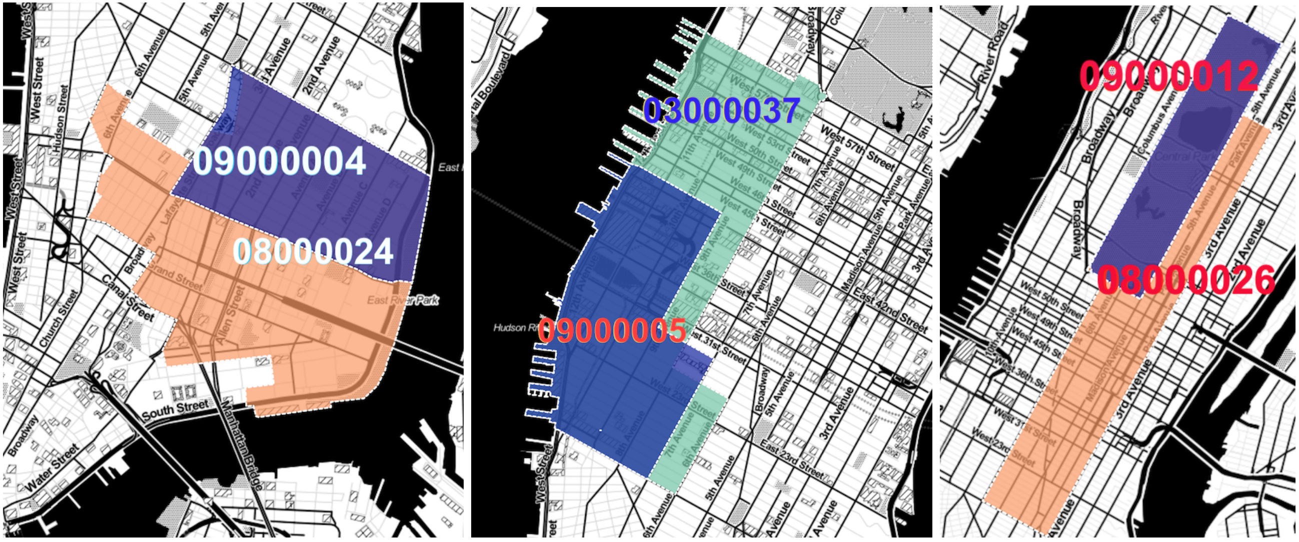
Category	*Score*	*Category*	*Score*
8. Shops & Services	0.312	5. Outdoors & Recreation	0.250
7. Residence	0.130	9. Travel & Transport	0.117
2. College & Education	0.114	6. Professional Services	0.071
4. Nightlife Spot	0.067	3. Food & Restaurant	0.052
1. Arts & Entertainment	0.008		

Table 8: NDCG@k Scores with chart on ROI Semantic Annotation Ranking Analysis on ROI pairs.



	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5
GE_POI	0.609	0.613	0.660	0.710	0.750
CrossMap	0.629	0.630	0.661	0.695	0.725
BiNE	0.650	0.644	0.669	0.696	0.728
TNE_WCR	0.727	0.671	0.685	0.717	0.741
TNE_NW	0.752	0.693	0.692	0.724	0.750
TNE	**0.844**	**0.783**	**0.794**	**0.831**	**0.854**

Lower East Manhattan, NYC **ROIs: 09000004,08000024**	**West Manhattan, NYC** **ROIs: 03000037,09000005**	**Midtown Manhattan, NYC** **ROIs: 09000012,08000026**
1. Arts & Entertainment	6. Professional Services	1. Arts & Entertainment
3. Food & Restaurant	9. Travel & Transport	2. College and Education
8. Residence	2. College & Education	8. Shops & Services



(a) Lower East Side. (b) West Side. (c) Midtown.

Fig. 9: Semantic difference on overlapped ROIs.

all categories with level 3 should rank higher than level 2, followed by level 1 categories at the bottom. We used Normalized Discounted Cumulative Gain (NDCG) [44] to find the quality of embedding via ranking order. Table 8 shows NDCG scores at top-k ranking positions, and higher the score signifies better ranking order achieved by the model. Result presented in Table 8 suggests that TNE beats all baselines GE_POI, CrossMap, BiNE, TNE_WCR and TNE_NW by a considerable margin. TNE achieved an NDCG@1 score of 0.844 and an average NDCG@k ($k = [1, 5]$) of 0.8206 with 9 semantic categories.

It is an important result in our experiment that gives us insights on how ROI embeddings can capture the semantic perspective observed by society about any region. From Table 8, we follow that TNE outperformed GE_POI, CrossMap and BiNE by 0.235, 0.215, 0.194 NDCG at rank 1 which is considerably high improvement in selecting the best category candidate for an ROI. The results are similar to other ranking levels. An average NDCG gain of more than 20% from state of the art baselines (i.e., GE_POI, CrossMap, and BiNE) is a large gain (in ranking problem) that shows the efficacy of TNE. Also, to note that TNE_NW and TNE_WCR performed better than other baselines but beaten by TNE with an average score of 0.1 (or 12%). It shows the necessity of using edge weights in $\mathcal{G}$, and community-aware random walk in our strategy.

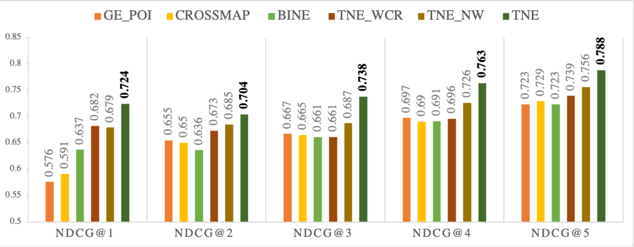
4.5 Semantic Category Difference from ROI Embeddings

In this section, we briefly demonstrate the capability of ROI embedding to find semantic differences between ROIs. Technically, for any pair of ROIs with embedding vectors $\vec{r_1}, \vec{r_2}$ and semantic category vectors $C = \{\vec{c_1}, \dots, \vec{c_9}\}$, the top semantic category difference is calculated as follows.

$$\max_{\{\vec{c}\in\mathcal{C}\}} \text{abs}(\langle \vec{d}, \vec{c} \rangle) \qquad \text{where } \vec{d} = \vec{r_1} - \vec{r_2}.$$

We demonstrate semantic category difference of 3 pairs of overlapped ROIs from lower east, west and midtown of Manhattan as shown in Figure 9. We ranked the top three semantic category differences for each pair with the formulation mentioned before. The result is presented in the Table within Figure 9 and on close observation it reveals discernible facts. The major semantic category differences between the pair of ROIs (09000004,08000024) from *Lower East Manhattan* shown in Figure 9a are *Arts & Entertainment* and *Residence* it is because ROI 09000004 has popular music and theater performance centers and has a large residential community known as East Village and on the contrary lower part ROI 08000024 shown in orange has many restaurants. Similarly, for the pair of ROIs (03000037,09000005) from west Manhattan shown in Figure 9b has major

Table 9: NDCG@k Scores for Semantic Category Difference Annotation on ROI pairs



	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5
GE_POI	0.576	0.655	0.667	0.697	0.723
CROSSMAP	0.591	0.650	0.665	0.690	0.729
BINE	0.637	0.636	0.661	0.691	0.723
TNE_WCR	0.682	0.673	0.661	0.696	0.739
TNE_NW	0.679	0.685	0.687	0.726	0.756
TNE	**0.724**	**0.704**	**0.738**	**0.763**	**0.788**

difference with *Travel & Transport* and *College & Education* since ROI 03000037 contains the transit hub of Manhattan (Port Authority) and universities such as The City University of New York and State University of New York and similar places does not feature in ROI 09000005. Lastly, the midtown Manhattan with ROIs (09000012,08000026) shown in Figure 9c does not show *Recreation* as major category difference as ROI 08000026 fully covers ROI 09000012 i.e. Central Park (Recreation place) and shows differences in *Arts & Entertainment, Shops & Services* as ROI 08000026 is a cultural hub and a shopping or commercial area.

We performed an in-depth study of the semantic category difference annotation with NDCG analysis, similar to the analysis in Section 4.4. We chose 30 pairs of ROIs, and human raters annotated all categories on each pair of ROIs in three levels based on their differences as *(1) non-significantly (2) moderately (3) critically* different. In an ideal case, the analysis from embedding should rank categories in the order, 3 *critically*, 2 *moderately*, and 1 *non-significantly* different categories. Table 9 shows the performance of each model on NDCG analysis. We still found TNE to perform better than other baselines.

4.6 Region Popularity Prediction

To evaluate the effectiveness of ROI embedding in a real-world application, we performed the popularity prediction of region experiment. We used an open available check-in dataset of New York City [49] for the prediction task. The only feature used for prediction is the ROI embedding obtained from baselines and TNE models. We used two regression models (a) Random Forest, and (b) XGBoost, for prediction of the number of check-in in a region. Table 10 shows the mean absolute error (MAE), and root mean squared error (RMSE) for both the regression models. We can notice that TNE performed well in comparison to baselines in all except XGBoost-MAE where TNE_NW performed best. However, the RMSE error for TNE_NW is very high for both regressions. We also performed a temporal (day, night) region popularity experiment, shown in Table 11. TNE_DAY, and TNE_NIGHT are TNE models trained with G_{rw} graph generated from geotagged tweets obtained during days and nights, respectively.

Table 10: Region Popularity Prediction

	Random Forest		XGBoost	
	MAE	RMSE	MAE	RMSE
GE_POI	855.77 ±39.4	1415.41 ±79.8	929.7	1731.65
CROSSMAP	674.85 ±16.7	1283.1 ±20.25	779.2	1429.92
BINE	636.89 ±14.52	1476.1 ±31.47	661.1	1415.37
TNE_WCR	661.53 ± 13.72	1471.17 ± 3.81	653.07	1498.37
TNE_NW	636.96 ± 8.07	1559.67 ± 6.88	**639.63**	1585.69
TNE	**633.73** ±**13.66**	**1228.5** ±**19.84**	646.01	**1228.57**

Table 11: Day and Night Region Popularity Prediction

	Random Forest MAE	
	Day Popularity	Night Popularity
TNE	673.49 ± 14.33	629.34 ± 18.06
TNE_DAY	**646.08** ± **11.53**	616.43 ± 10.54
TNE_NIGHT	667.81 ± 13.70	**611.62** ± **12.31**

Summary: Each experiment investigates a qualitative aspect of the embedding procedures. TNE provides a qualitative semantic embedding, shown via semantic category annotation experiments. The spatial affinity experiment exhibits that TNE preserves strong geospatial relations. Region popularity prediction with embedding features demonstrates the expressiveness of features from the models. From all the above experiments it can be established that our approach for ROI embeddings with TNE shows admissible support on the quality of ROI representation.

5 Related works

To the best of our knowledge, only one very recent work by Jenkins et al. [22], builds an ROI embedding jointly with rich auxiliary information – in their case POI, satellite images, and taxi flow data. While our approach also uses POI data, it makes the (as we see very important) distinction to weight these by popularity, and also equally incorporates semantic information from microblog text. This allows for a different and (in our view) richer set of applications demonstrated, including temporal variation using timing of microblog updates. The embedding methods are also different, while their approach uses a single auto-encoder from a convolutional network, we show how to build a tripartite network that can ensure the three components (ROIs, POIs, and semantic text) can be weighted equally. Although this work is only in-press, and their data is private, we still attempt to compare against this method by considering similar baselines – noteably including method TNE_NW which like Jenkins et al. [22] does not include popularity weights on POIs in the G_{pw} graph.

POI Embedding. Extended literature survey suggests that research work on *Places of Interest* (POI) embedding are the most closest studies to our work. But there are major differences in our ROI embedding from the works on POI embedding [18, 39, 43, 47, 48, 50, 54]. *Firstly,* our work treats ROI as considerably bigger regions encircling many POIs, simply aggregating POI embedding vectors to generate ROI embedding does not yield desired result, as we will see in our comparative experiments. *Secondly,* relevant POI embedding learning works focuses on POI sequence recommendation task for users based on checkin activity [18, 43, 48, 50], whereas, our task on ROI embedding focuses on preserving spatial and semantic relation without involving users in the scenario. That makes our problem statement different from others. *Thirdly,* POI embedding work by Xie et al. [47] modeled a bipartite graph network embedding for learning POI which also consist of a *POI-Region* bipartite graph. Though the concept of region is unclear from their paper, we assumed our definition for ROI for a comparative analysis. Major difference in our work is that we capture the social behaviour within region and also transitive/implicit relationship for bipartite graphs. Since, POI recommendation task is extraneous to our problem statement we cannot directly compare their task/experiment with ours. *Fourthly,* the work of Zhang et. al. [54] aims to find correlation among *hotspot locations* (defined as spatial gaussian kernel window), word and time to search spatio-temporal events. We find our work dissimilar from [54] as *hotspot locations* are very different from our geographically bounded polygonal ROI or POIs. Both of these spatial entity plays significantly different role in our model.

Semantic-Visual Embedding. The idea of cross-modal embedding in one-shot supervised learning has recently garnered researchers' attention. From the bird-eye view, we find our objective moderately matches semantic-visualization embedding on images where the problem is the assignment of semantic labels on sub-region/partial image [13, 38]. Our semantic embedding of ROIs also uses multimodal features to find the uniqueness of a spatial region. However, there are distinctions between the two fields of work. Our work's novelty lies in the application of semantic features on the real-world geospatial regions of interest (ROIs) from the perspective of social engagement and solving the specific problems related to it. Additionally, the former focuses on feature-based spatial search on images, whereas our work concentrates on relational-based semantic learning on graph networks. In that aspect, our work is entirely original in the geospatial domain.

Graph Network Embedding. Broadly our work is related to *Network Embedding* research. The commonly used methods for network embedding are matrix factorization, random walk, deep neural networks. Our model is based on random-walk and Deepwalk [35] is the first pioneer work on it. We made advancements in the field with structure-preserving tripartite or multipartite network embedding following the footsteps after groundbreaking contribution from LINE, HINE, Metapath2vec++, PME, BiNE, etc. [5, 7, 12, 16, 40].

The first use of *tri-party* or *three entity* in graph network embedding in alignment with random-walk strategy is from Pan et al. [33]. However, it is not a true *tripartite graph network*, rather an attributed heterogeneous embedding approach involving text associated-entity by incorporating contextual word embedding. Another more closely related work on *tripartite embedding* is HGP from Kim et al. [24] involving *group→user→item* and does not consider *group-item* relationship in the picture which does not make it a complete tripartite network. HGP propagates relation for each edge type independently, and their approach concentrates on attention mechanisms for large scale adaptation. Overall, the main aim of HGP [24] is to tackle the oversmoothing problem in heterogeneous graphs on a large scale, which is very different from our objective of incorporating implicit and explicit relationship in learning representation.

More recent work from Hong et al. [20] aligns their research direction towards attributed network embedding in a different direction. Each vertex in the graph network has a fixed set of features to evaluate their similarity. While these works [20, 24, 33] mainly concentrate on feature-attributed network embedding, our work focuses on capturing implicit structural information from transitive relations on multipartite graph networks.

Furthermore, as described in BiNE [16], the random walk generator used in the works mentioned above (inspired by [35]) are not equipped to mimic the real-world distribution of vertices in a graph. BiNE [16] overshadows them in structure-preserving embedding, which thoroughly investigates vital information on edge relationship in graph network along with the oversmoothing problem of vertices. Hence, we also add BiNE [16] as a baseline for our experiments, where our model proposes community-aware random walk, transitive property preserving graphs, and a heterogeneous negative sampling technique for multiple entities embedding.

We thank reviewers of this paper to bring a very recent work of Chen et al. [4] to our notice, which explores *folded* bipartite network embedding using graph convolution network (GCN). This work advances bipartite network embedding by introducing higher-order relationships and using a self-attention technique to perform embedding. Our work concentrates on extending bipartite to multipartite network embedding with random walk modeling and supporting our use-case with a real-world application, which makes [4] partially orthogonal.

We believe our work contributes significantly towards structure-preserving network embedding and its application in semantic ROI embedding to herald a new direction in elucidating geospatial regions with semantic features.

6 Conclusion

In this paper, we propose TNE, a tripartite network embedding model for learning *Regions of Interest* (ROI) embedding. Our study focuses on learning ROI embedding that simultaneously captures semantic and geospatial features. First, we formalize the semantic embedding for ROIs problem with an information graph that captures social, semantic, and spatial attributes. Then we use that TNE induces transitive relational features to obtain better learning performances while preserving the structure of the information graph. We performed multifaceted experiments on real-world data showing the advantages of performing ROI embedding with TNE over other baselines. Also, we demonstrate an interactive map to explore and discover the similarities and distinctness of regions.

Acknowledgment We thank our colleagues who helped in crowdsourcing dataset for experiments. We would also like to show our gratitude to Sunipa Dev for early discussion on this topic, and initial help in processing data.

References

1. Alzahrani, T., Horadam, K.J., Boztas, S.: Community detection in bipartite networks using random walks. In: Complex Networks V. Springer (2014)
2. ArcGIS: Arcgis.com (2019). URL `https://arcgis.com/`. [Online; accessed 15-March-2019]
3. Buyukokkten, O., Cho, J., Garcia-Molina, H., Gravano, L., Shivakumar, N.: Exploiting geographical location information of web pages. ilpubs.stanford.edu (1999)
4. Chen, H., Yin, H., Chen, T., Wang, W., Li, X., Hu, X.: Social boosted recommendation with folded bipartite network embedding. IEEE Transactions on Knowledge and Data Engineering (2020)
5. Chen, H., Yin, H., Wang, W., Wang, H., Nguyen, Q.V.H., Li, X.: Pme: projected metric embedding on heterogeneous networks for link prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1177–1186 (2018)
6. Chen, L., Yuan, F., Jose, J.M., Zhang, W.: Improving negative sampling for word representation using self-embedded features. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 99–107. ACM (2018)
7. Chen, Y., Wang, C.: Hine: Heterogeneous information network embedding. In: International Conference on Database Systems for Advanced Applications, pp. 180–195. Springer (2017)
8. Cheng, T.K., Von Behren, J.R.: Location-based searching using a search area that corresponds to a geographical location of a computing device (2013). US Patent 8,386,514
9. CityOfNewYork: cityofnewyork (2019). URL `https://opendata.cityofnewyork.us/`. [Online; accessed 15-March-2019]
10. Courbariaux, M., Bengio, Y., David, J.P.: Binaryconnect: Training deep neural networks with binary weights during propagations. In: Advances in neural information processing systems, pp. 3123–3131 (2015)
11. Deng, H., Lyu, M.R., King, I.: A generalized co-hits algorithm and its application to bipartite graphs. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 239–248. ACM (2009)
12. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 135–144. ACM (2017)
13. Engilberge, M., Chevallier, L., Pérez, P., Cord, M.: Deep semantic-visual embedding with localization (2018)
14. Foursquare: foursquare.com (2019). URL `http://foursquare.com`. [Online; accessed 15-March-2019]
15. Fuglede, B., Topsoe, F.: Jensen-shannon divergence and hilbert space embedding. In: Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on, p. 31. IEEE (2004)
16. Gao, M., Chen, L., He, X., Zhou, A.: Bine: Bipartite network embedding. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 715–724. ACM (2018)
17. H. Butler, M.Daly, A. Doyle, S. Gillies, S. Hagen, T.Schaub: The GeoJSON Format. RFC 7946, RFC Editor (2016). DOI 10.17487/RFC7946. URL `http://www.rfc-editor.org/rfc/rfc7946.txt`

18. He, T., Yin, H., Chen, Z., Zhou, X., Sadiq, S., Luo, B.: A spatial-temporal topic model for the semantic annotation of pois in lbsns. ACM Transactions on Intelligent Systems and Technology (TIST) **8**(1), 12 (2016)
19. Hellinger, E.: Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. Journal für die reine und angewandte Mathematik **136**, 210–271 (1909)
20. Hong, R., He, Y., Wu, L., Ge, Y., Wu, X.: Deep attributed network embedding by preserving structure and attribute information. IEEE Transactions on Systems, Man, and Cybernetics: Systems (2019)
21. Im, D.J., Verma, N., Branson, K.: Stochastic neighbor embedding under f-divergences. arXiv preprint arXiv:1811.01247 (2018)
22. Jenkins, P., Farag, A., Wang, S., Li, Z.: Unsupervised representation learning of spatial data via multimodal embedding. In: CIKM (2019). DOI 10.1145/3357384.3358001
23. Jiang, M., Cui, P., Yuan, N.J., Xie, X., Yang, S.: Little is much: Bridging cross-platform behaviors through overlapped crowds. In: AAAI, pp. 13–19 (2016)
24. Kim, K.M., Kwak, D., Kwak, H., Park, Y.J., Sim, S., Cho, J.H., Kim, M., Kwon, J., Sung, N., Ha, J.W.: Tripartite heterogeneous graph propagation for large-scale social recommendation. arXiv preprint arXiv:1908.02569 (2019)
25. Kleinberg, J.M.: Hubs, authorities, and communities. ACM computing surveys (CSUR) **31**(4es), 5 (1999)
26. Kliman-Silver, C., Hannak, A., Lazer, D., Wilson, C., Mislove, A.: Location, location, location: The impact of geolocation on web search personalization. In: Proceedings of the 2015 Internet Measurement Conference, pp. 121–127. ACM (2015)
27. Lee, J.A., Renard, E., Bernard, G., Dupont, P., Verleysen, M.: Type 1 and 2 mixtures of kullback–leibler divergences as cost functions in dimensionality reduction based on similarity preservation. Neurocomputing **112**, 92–108 (2013)
28. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
29. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013)
30. NPS: Statue of liberty, park statistics (2019). URL `https://www.nps.gov/stli/learn/management/park-statistics.htm`. [Online; accessed 11-July-2019]
31. NYCDataBeta: Nyc geojson dataset (2019). URL `\url{http://data.beta.nyc/dataset\?res_format=GeoJSON}`. [Online; accessed 15-Mar-2019]
32. nyc.gov: Nyc open data (2019). URL `\url{https://opendata.cityofnewyork.us/}`. [Online; accessed 15-Mar-2019]
33. Pan, S., Wu, J., Zhu, X., Zhang, C., Wang, Y.: Tri-party deep network representation. Network **11**(9), 12 (2016)
34. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543 (2014)
35. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701–710. ACM (2014)
36. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the national academy of sciences **101**(9), 2658–2663 (2004)
37. Raymond, J.W., Gardiner, E.J., Willett, P.: Rascal: Calculation of graph similarity using maximum common edge subgraphs. The Computer Journal **45**(6), 631–644 (2002)
38. Ren, Z., Jin, H., Lin, Z., Fang, C., Yuille, A.: Multi-instance visual-semantic embedding. arXiv preprint arXiv:1512.06963 (2015)
39. Shen, J., Cheng, T.: Semantic enrichment of interesting regions with poi data. In: Proceedings of the GISRUK conference, Manchester, pp. 1–5 (2017)
40. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web, pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
41. Thearling, K.: An introduction to data mining. Direct Marketing Magazine pp. 28–31 (1999)
42. Twitter: twitter.com (2019). URL `https://twitter.com`. [Online; accessed 15-June-2019]
43. Wang, Y., Qin, Z., Pang, J., Zhang, Y., Xin, J.: Semantic annotation for places in lbsn through graph embedding. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 2343–2346. ACM (2017)
44. Wang, Y., Wang, L., Li, Y., He, D., Liu, T.Y.: A theoretical analysis of ndcg type ranking measures. In: Conference on Learning Theory, pp. 25–54 (2013)
45. Wikipedia: Region of interest, wikipedia, the free encyclopedia (2019). URL `https://en.wikipedia.org/wiki/Region\_of\_interest`. [Online; accessed 11-July-2019]
46. Wikipedia: Wikipedia, the free encyclopedia (2019). URL `https://www.wikipedia.org/`. [Online; accessed 15-June-2019]
47. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based poi embedding for location-based recommendation. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 15–24. ACM (2016)
48. Xie, M., Yin, H., Xu, F., Wang, H., Zhou, X.: Graph-based metric embedding for next poi recommendation. In: International Conference on Web Information Systems Engineering, pp. 207–222. Springer (2016)
49. Yang, D., Zhang, D., Zheng, V.W., Yu, Z.: Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. IEEE Transactions on Systems, Man, and Cybernetics: Systems **45**(1), 129–142 (2015)
50. Yin, H., Wang, W., Wang, H., Chen, L., Zhou, X.: Spatial-aware hierarchical collaborative deep learning for poi recommendation. IEEE Transactions on Knowledge and Data Engineering **29**(11), 2537–2551 (2017)
51. Yu, L., Zhang, C., Pei, S., Sun, G., Zhang, X.: Walkranker: A unified pairwise ranking model with multiple relations for item recommendation. In: AAAI. AAAI (2018)
52. Yule, G.U.: On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship. Biometrika **30**(3/4), 363–390 (1939)
53. Zager, L.A., Verghese, G.C.: Graph similarity scoring and matching. Applied mathematics letters **21**(1), 86–94 (2008)

54. Zhang, C., Zhang, K., Yuan, Q., Peng, H., Zheng, Y., Hanratty, T., Wang, S., Han, J.: Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning. In: Proceedings of the 26th International Conference on World Wide Web, pp. 361–370. International World Wide Web Conferences Steering Committee (2017)
55. Zhang, Z., Zweigenbaum, P.: Gneg: Graph-based negative sampling for word2vec. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, pp. 566–571 (2018)
56. Zhao, S., Zhao, T., King, I., Lyu, M.R.: Gt-seer: geo-temporal sequential embedding rank for point-of-interest recommendation. arXiv preprint arXiv:1606.05859 (2016)