Unsupervised Learning of Parsimonious General-Purpose Embeddings for User and Location Modeling

JING YANG and CARSTEN EICKHOFF, ETH Zurich

Many social network applications depend on robust representations of spatio-temporal data. In this work, we present an embedding model based on feed-forward neural networks which transforms social media check-ins into dense feature vectors encoding geographic, temporal, and functional aspects for modeling places, neighborhoods, and users. We employ the embedding model in a variety of applications including *location recommendation*, *urban functional zone study*, and *crime prediction*. For *location recommendation*, we propose a Spatio-Temporal Embedding Similarity algorithm (STES) based on the embedding model.

In a range of experiments on real life data collected from Foursquare, we demonstrate our model's effectiveness at characterizing places and people and its applicability in aforementioned problem domains. Finally, we select eight major cities around the globe and verify the robustness and generality of our model by porting pre-trained models from one city to another, thereby alleviating the need for costly local training.

CCS Concepts: • **Networks** → Location-based services; • **Information systems** → Recommender systems; • **Human-centered computing** → Collaborative and social computing;

Additional Key Words and Phrases: Social networks, check-in embedding, personalized location recommendation, urban functional zone study, crime prediction

ACM Reference format:
Jing Yang and Carsten Eickhoff. 2018. Unsupervised Learning of Parsimonious General-Purpose Embeddings for User and Location Modeling. *ACM Trans. Inf. Syst.* 36, 3, Article 32 (March 2018), 33 pages.
https://doi.org/10.1145/3182165

1 INTRODUCTION

Spatial social network applications and services, such as transport network design, location-based services, urban structure learning, and crime prediction, normally involve two important issues: understanding residents' real-time activities and accurately describing urban spaces [6, 9, 50]. Toward the former, researchers usually rely on check-in data from social network platforms (e.g., Twitter, Foursquare, and Facebook) or specifically designed surveys, which cover a significant number of users in the form of check-ins and comments about points-of-interest (POIs). For the latter aspect, place annotation approaches are required.

Among approaches to annotate places, a common method is to simply employ categorical labels such as *Home*, *Restaurant*, and *Shop* [13, 22, 49, 67]. Although straightforward, it is unclear whether such discrete tags offer sufficient flexibility and descriptive power for modeling the complexity of urban landscapes.

Authors' addresses: J. Yang, ETH Zurich, Raemistrasse 101, Zurich, 8092, Switzerland; email: jing.yang@inf.ethz.ch; C. Eickhoff, Center for Biomedical Informatics, Brown University, Providence, RI, 02912, U.S.A; email: carsten@brown.edu.

https://doi.org/10.1145/3182165

In this work, we instead represent places by means of embedding vectors in a semantic space. Aiming to annotate places in terms of temporal, geographic, and functional aspects, we extract the time, location, and venue function from check-in records and train our model in the context of check-in sequences which originate from an individual user or neighborhood.

In comparison with the traditional discrete method, our approach describes places in a continuous manner and preserves more information about people's real behavior as well as places' day-to-day usage patterns. For instance, in the case of label annotation, three food related places which serve Chinese breakfast, pizza, and sushi, respectively, may all be labeled as *Restaurant* but their features in food type, active hours, and location may vary dramatically. In the course of this article, we will show how our embedding model represents such within-class variance in a natural way. As embedding vectors are learned from people's real-time check-ins, we also leverage them in user representation to reflect people's activity patterns and interests.

The embedding model is an accurate descriptor of places and users in terms of geographic and functional affinity, activity preferences, and daily schedules. As a consequence, it can be applied in a wide range of settings. In this article, we consider three practical applications: *location recommendation*, *urban functional zone study*, and *crime prediction*. Our empirical investigation is driven by five research questions:

- —RQ1. How well does the embedding model differentiate locations and users along temporal, geographic, and functional aspects?
- —RQ2. How does our location recommendation algorithm STES compare to state-of-the-art methods?
- —RQ3. How to define and visualize urban functional zones using the proposed model?
- —RQ4. How well can the model predict typical urban characteristics?
- —RQ5. With what generalization error can an embedding model trained in one city be transferred to other cities?

By answering these research questions, we make three novel contributions:

- —We present an **unsupervised** spatio-temporal embedding scheme based on social media check-ins. Trained with monthly check-in sequences, the model shows wide applicability for tasks ranging from social science problems to personalized recommendations.
- —Based on this model, we propose the **STES** algorithm that recommends locations to users. Compared with state-of-the-art recommendation frameworks, we can achieve an improvement up to 30%.
- —The model shows strong **robustness** and **generality**. Once trained in a representative city, it can be directly utilized in other cities with only slight generalization errors ($< 3\%$).

The remainder of this article is structured as follows: Section 2 summarizes existing literature on check-in embedding learning as well as state-of-the-art works in *location recommendation*, *urban functional zone study*, and *crime prediction*. Section 3 describes our embedding model and the STES location recommendation algorithm. Section 4 empirically evaluates the performance of our model on a number of tasks, comparing to a range of competitive performance baselines. Finally, Section 5 concludes with a summary of our findings and a discussion of future work.

2 RELATED WORK

In this section, we first discuss the progress of place annotation using check-in data. We then review embedding learning techniques and their applications on the basis of social networks. Finally, we introduce the state-of-the-art in our three application domains.

2.1 Place Annotation Using Check-in Data

Among place annotation works that involve social media or survey data, most studies utilize existing venue category tags and formulate the problem as a prediction or clustering task. Sarda et al. [49] propose a spatial kernel density estimation based model on the basis of the 2012 Nokia Mobile Data Challenge (MDC) [24] to label an unknown place with one of 10 semantic tags (e.g., *Home*, *Transport Related*, *Shop*). He et al. [13] design a topic model framework which takes user check-in records as input and annotates POIs with category-related tags from Twitter and Foursquare. Noulas et al. [41] represent areas by counts of inner check-in venue categories and further implement clustering. Krumm and Rouhana [22] propose an algorithm to classify locations into 12 labels (e.g., *School*, *Work*, *Recreation Spots*) based on visitor demographics, time of visit, and nearby businesses of the locations. Ye et al. [67] propose a support vector machine based algorithm to annotate places with 21 category tags. As mentioned earlier, such discrete usage of category labels carries insufficient descriptive power for modeling users and the real activities in urban spaces. Therefore, a continuous vector representation in semantic space is proposed in our work.

2.2 Embedding Learning Techniques and Applications

In recent years, due to promising performance in capturing the contextual correlation of items, approaches to embedding items in Euclidean spaces have become popular and have been applied in a variety of domains, including E-commerce product recommendation [44], network classification [43], user profiling [52, 53], an so forth. Social media contexts are active fields of application. Wijeratne et al. [63] embed words in tweet texts and twitter profile descriptions into vectors for gang member identification. Tang et al. [54] learn embeddings of sentiment-specific words in tweets for sentiment classification. Lin et al. [30] develop a matrix sentence embedding framework and adopt it in Yelp reviews for user sentiment analysis. In these cases, however, embedding techniques are straightforward applications to textual documents. In our work, we develop an analogous situation by treating check-in sequences as virtual sentences. Consequently, correlations of contextual locations and activities can be better modeled.

2.3 Location Recommendation

As the most common performance benchmark for spatial models [3, 68], location recommendation has been popular in recent years in studies on location-based social networks (LBSNs). The most basic location recommendation approaches are content based, relying solely on properties of users and locations [35, 55]. The idea is to explore user and location features, then make recommendations based on their similarities and past preferences. Another popular branch of approaches employs matrix factorization (MF) [21, 48] and its derivative methods [3, 4, 29, 31]. MF-related methods aim to represent users and items in matrices and recommend locations based on row-to-row correlations. Methods based on topic models (TMs) [2] and Markov models (MMs) [36] also perform well in recommendation tasks. Here, geographic and temporal information are often included as additional evidence [8, 15, 17, 23, 28, 75]. There exist several topic-model based location recommendation methods which consider geographic, temporal, and even venue-specific semantic information [58, 59, 69, 70, 72, 73]. However, their problem setting is different. Instead of predicting the next place which a user might visit, they focus on recommending a nearby venue where a user has never been before. To optimize such a recommendation, they further introduce the concepts of *home users* and *visitor users* to accommodate for various visitation constellations.

Recently, some recommendation methods [33, 42, 76] involve neural network embedding techniques in their processing schemes. However, they only focus on geographic location modeling but ignore other information (e.g., venue function, check-in time) associated with each check-in.

Consequently, these models hold very limited applicability beyond the immediate context of location recommendation that they were designed for. There also exist some recommendation approaches involving Long Short-Term Memory (LSTM) recurrent neural networks [25, 51]. However, such methods fall into the class of supervised learning, requiring costly annotations that are not always available. A recent related work proposed a spatial-aware hierarchical collaborative deep learning model for location recommendation [71] which demonstrates promising performance in handling the cold start issue. However, like the works of topic modeling approaches above, the problem scenario is to recommend POIs which have never been visited before. Additionally, these methods exploit the semantic representation of POIs in a supervised and task-guided way. In contrast, we develop an unsupervised learning approach including various types of check-in information which can be utilized in a wide range of settings by giving a robust representation of each place without requiring supervised label information. Section 4 will highlight the resulting performance differences both in the context of recommendation as well as other tasks.

2.4 Urban Functional Zone Study

Usually, urban functional zone partitioning is purely based on geographic data from Geographic Information System (GIS) and government datasets [56, 66]. In recent years, researchers began to incorporate crowdsourced activity data from social media, public surveys, and traces of personal mobility into such studies. Xing et al. [65] use mobile billing data to cluster urban traffic zones. Yuan et al. [74] design a topic model and use POIs and taxi trajectories together with public transit data to cluster neighborhoods in Beijing into six functional zones such as *Education*, *Residence*, and *Entertainment*. Zhu et al. [77] leverage 2014 Puget Sound travel survey data,[1] Foursquare POIs, and Twitter temporal features to characterize and cluster neighborhoods into four functional areas: *Shopping*, *Work*, *Residence*, and *Mixed functionalities*. Cranshaw et al. [7] propose a spectral clustering method to directly group POIs based on their check-in information. In our work, we simply utilize the check-in vectors trained by our embedding model to characterize neighborhoods and further cluster urban functional zones. We show that our method is capable of encoding people's daily activity patterns to discover the true underlying usage of urban spaces.

2.5 Crime Prediction

Crime prediction is a common social science issue. Iqbal et al. [18] rely on demographic features such as population, household income, and education level as input to predict regional crime rates on a three-point scale. Gerber [9] and Wang et al. [60] design topic models based on Twitter posts to predict criminal incidences. In this work, we demonstrate that people's day-to-day activity patterns are strong indicators of crime occurrence. Based on the proposed embedding model, we characterize neighborhoods with check-in vectors. Acting as features, these vectors perform well in future crime rate and crime occurrence prediction.

3 METHODOLOGY

In this section, we begin by introducing the problem domain. Then, we describe the check-in embedding model. Finally, we propose the model-based location recommendation algorithm STES.

3.1 Scenario

We first process social media check-ins in terms of their temporal, geographic, and functional aspects. Important definitions are proposed as follows.

[1] http://www.psrc.org/data/transportation/travel-surveys/2014-household.

Table 1. Time Discretization

Timeslot Tag	Time
Morning/WeekendMorning	6:00AM–10:59AM
Noon/WeekendNoon	11:00AM–1:59PM
Afternoon/WeekendAfternoon	2:00PM–5:59PM
Evening/WeekendEvening	6:00PM–9:59PM
Night/WeekendNight	10:00PM–5:59AM

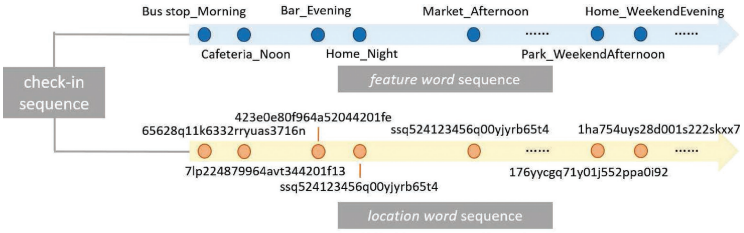


Fig. 1. A check-in sequence is composed of a *feature word* sequence and a *location word* sequence. Each dot represents a unique check-in. The dots are distributed unevenly to reflect the varying time spans between check-ins.

Check-ins. A check-in is defined as a tuple $c =< u, f, t, l >$ which depicts that a user u visits a location l at time t, where f demonstrates the functional role of the visited venue. To process the raw time data t, motivated by a natural reflection of daily routines and the dataset density, we discretize t in 10 functional timeslots, namely, *Morning*, *Noon*, *Afternoon*, *Evening*, *Night*, *WeekendMorning*, *WeekendNoon*, *WeekendAfternoon*, *WeekendEvening*, and *WeekendNight*. Discretization thresholds are listed in Table 1. *Morning* is the time bucket when people start a day and work before lunch. *Noon* is the lunch break time. *Afternoon* refers to working hours after lunch. *Evening* is for dinner and after-work activities. *Night* corresponds to the sleeping hours. As daily activities on weekends are usually different from those on weekdays (e.g., *WeekendAfternoon* is often for leisure instead of work on weekday *Afternoon*), we additionally define a set of timestamps on weekends that has the same temporal correspondence. To represent a check-in's functional role f, we utilize the popular Foursquare hierarchy of venue categories.[2] As for the location l, we leverage the unique ID of each place.

For each check-in, only its function f, time t, and location l are combined and trained to obtain the embedding vectors. There exist around 400 functional roles, 10 timeslots, and thousands of unique locations. Due to the large amount of locations, simply concatenating all these three aspects into one check-in word would result in a relatively sparse distribution of data points given the scale of available data. Therefore, to achieve a good balance, for each check-in record, we concatenate functional role f and timeslot t as its *feature word* (e.g., *"Bar_Evening"*), and use the unique ID of location l as its *location word* (e.g.,*"423e0e80f964a52044201fe3"*).

Check-In Sequences. Check-in sequences are described by two parallel sub-sequences: a *feature word* sequence and a *location word* sequence. Words in these two sequences are one-to-one correspondent. Both the *feature word* sequence and *location word* sequence are chronological orderings of check-ins in 1 month of the profile of a user u or a neighborhood n. Figure 1 shows a check-in sequence example.

[2]https://developer.foursquare.com/categorytree.

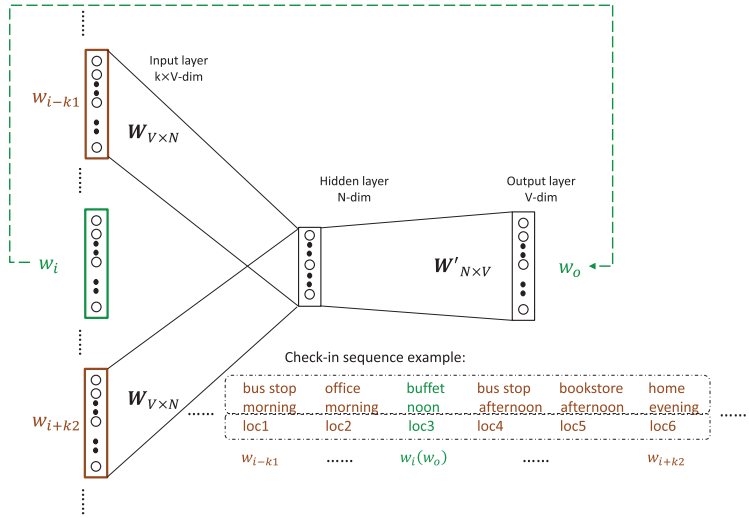


Fig. 2. The embedding training framework. Each target word (highlighted in green) takes turns to be predicted from its context words (highlighted in brown).

Users/Neighborhoods. Depending on whether the task is user-centric (location recommendation) or area-centric (urban functional zone study and crime prediction), a user u or a neighborhood n is taken as the context from which check-in sequences are extracted.

Given *feature word* sequences and *location word* sequences, our model learns embedding vectors of *feature words* and *location words* independently in two semantic spaces. Based on these intermediate representations, we calculate the vectors corresponding to check-ins, users, venues, and neighborhoods. This aggregation process will be elaborated in following sections.

3.2 Embedding Model

Figure 2 illustrates the neural network (NN) training framework. In the spirit of [39], each target check-in word (*feature word* or *location word*) $w_i(w_o)$ is predicted from preceding and following check-ins in a sliding window from w_{i-k1} to w_{i+k2}, where $k1$ and $k2$ are adjustable and $k = k1 + k2$ is the overall context window size. All check-in words are initialized with one-hot encoded vectors, which means for a given word, only one out of V vector components will be 1 while all others are 0. When the training process is finished, the row of weight matrix $\mathbf{W}_{V\times N}$ from input layer to hidden layer is the vector representation of the corresponding word.

The training objective is to minimize the loss function

$$E = -log\, p(w_o|w_{i-k1}, \ldots, w_{i+k2}). \tag{1}$$

where $p(w_o|w_{i-k1}, \ldots, w_{i+k2})$ is the probability of the target check-in given the context check-ins, which can be formulated as a softmax function

$$p(w_o|w_{ik}) = \frac{\exp({\mathbf{v}'_{w_o}}^T \mathbf{v}_{w_{ik}})}{\sum_{j=1}^{V} \exp({\mathbf{v}'_{w_j}}^T \mathbf{v}_{w_{ik}})}, \tag{2}$$

where

$$\mathbf{v}_{w_{ik}} = \frac{1}{k}(\mathbf{v}_{w_{i-k1}} + \cdots + \mathbf{v}_{w_{i+k2}}). \tag{3}$$

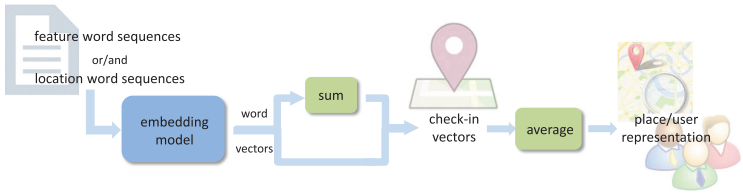


Fig. 3. Flowchart from importing check-ins to obtaining embedding vectors of places or users.

In Equation (2), $\mathbf{v}'_w$s comes from the columns of $\mathbf{W}'_{N\times V}$, the weight matrix connecting hidden layer to output layer. Back-propagation is applied during the training process and both hidden-to-output weights ($\mathbf{W}'$) and input-to-hidden weights ($\mathbf{W}$) are updated using stochastic gradient descent.

Also, from Equation (2), we can see that the learning process involves a traversal of all *feature words* or *location words*, which may jeopardize model efficiency. A typical method to tackle this problem is to employ the hierarchical softmax algorithm as proposed in [38]. To do so, we construct a Huffman binary tree [19], in which V vocabulary words are leaf units, and for each of them, there exists a unique path to the root. We only consider the along-path words when calculating the loss function. Our preliminary experiments demonstrate that utilizing hierarchical softmax improves the time efficiency by around 13%. On the other hand, the location recommendation accuracies after using hierarchical softmax are in most cases comparable with the raw results. The largest loss is approximately 0.7%.

Feature words and *location words* are separately trained via this model, resulting in a feature embedding space and a geographic embedding space. Now we can represent a check-in c by only its *feature word* vector or only its *location word* vector. Instead, to obtain a single joint representation, we follow [12] in summing up its *feature word* and *location word* vectors in element-wise fashion.

Furthermore, a user u can be represented by the mean of his/her check-in vectors, which also works if we only want to profile their activities in a specific time window. The same approach is applied to annotate a place or a neighborhood. Figure 3 demonstrates the entire workflow.

3.3 Recommendation Algorithm

Our recommendation algorithm is based on the user-location cosine similarity in the newly established embedding space. Recall that in the literature review, we mentioned how both temporal and geographic elements play important roles in recommendation tasks. Therefore, we utilize both *feature word* vectors ($\mathbf{v}_{fw}$) and *location word* vectors ($\mathbf{v}_{gw}$) to make recommendations. In this case, a check-in ($\mathbf{v}_c$) is represented by the element-wise summation of these two vectors

$$\mathbf{v}_c = \mathbf{v}_{fw} + \mathbf{v}_{gw}. \tag{4}$$

Such element-wise summation of feature vectors from different spaces has been successfully implemented in the field of computer vision [12] when training deep neural networks. As illustrated in Figure 4, similar to averaging, summation fuses features but without applying a re-scaling constant, which better preserves the information carried by the original feature vectors. In our preliminary experiments, we examined the location recommendation accuracy utilizing the check-in vectors calculated by summation, averaging, and concatenation, respectively. The results clearly indicate that summation-based vectors outperform the alternatives by 4–5% absolute performance.

On top of these summation-based vector representations, we profile locations and users. Remember that functional roles are defined by social network venue categories. For the sake of clarity, we will refer to "venue category" in place of "functional role" in the remainder of this section.

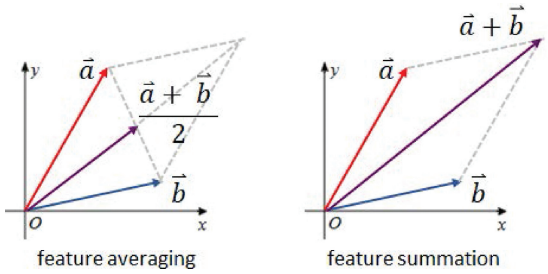


Fig. 4. Both averaging and summation fuse constituent vectors into a single final representation. Without the re-scaling constant, summation better preserves the information carried by the original feature vectors.

Table 2. Top 3 Most Visited Venue Categories in Different Timeslots

Timeslot	Top 3 popular venue categories
Morning	*Professional&Other places* (25.9%), *Food* (23.4%), *Travel&Transport* (21.9%)
Noon	*Food* (40.2%), *Professional&Other Places* (17.1%), *Shop&Service* (11.6%)
Afternoon	*Food* (27.9%), *Travel&Transport* (15.2%), *Shop&Service* (14.1%)
Evening	*Food* (30.3%), *Nightlife Spots* (19.5%), *Arts&Entertainment* (14.3%)
Night	*Nightlife Spots* (35.2%), *Food* (22.6%), *Travel&Transport* (11.7%)
WeekendMorning	*Food* (26.8%), *Outdoors&Recreation* (21.4%), *Travel&Transport* (18.9%)
WeekendNoon	*Food* (36.1%), *Outdoors&Recreation* (16.0%), *Shop&Service* (14.3%)
WeekendAfternoon	*Food* (30.1%), *Outdoors&Recreation* (16.7%), *Shop&Service* (15.8%)
WeekendEvening	*Food* (33.6%), *Nightlife Spots* (16.6%), *Arts&Entertainment* (14.0%)
WeekendNight	*Nightlife Spots* (49.4%), *Food* (21.1%), *Arts&Entertainment* (8.4%)

Location Profile. Although two locations may belong to the same venue category, they can still be differentiated if they are usually visited in different timeslots. A location l can thus be represented as $\mathbf{v}_l$ by averaging all user check-ins $\mathbf{v}_c$ issued there:

$$\mathbf{v}_l = \frac{1}{M}\sum_{m=1}^{M}\mathbf{v}_{c_m}, \tag{5}$$

where M is the total count of check-ins originating from location l.

User Profile. In a preliminary study, we confirm Li et al.'s hypothesis of check-in distributions differing across timeslots [27] (see Table 2). Correspondingly, and following intuition, frequently visited places vary according to time-of-day. Inspired by this observation, we calculate 10 profiles for each user corresponding to different timeslots. In each timeslot t, we represent a user u as $\mathbf{v}_{u_t}$ by averaging all his/her check-ins $\mathbf{v}_{c_t}$ in this timeslot and calculate a user coordinate centroid ($coordinate_{u_t}$) from those check-in locations ($coordinate_{c_t}$).

$$\mathbf{v}_{u_t} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{v}_{c_{t,n}}, \tag{6}$$

$$coordinate_{u_t} = \frac{1}{N}\sum_{n=1}^{N} coordinate_{c_{t,n}}, \tag{7}$$

where N is the total count of check-ins from user u in timeslot t.

Next, we calculate two *cosine similarities* for users in *each* timeslot. The first one, *user-activity similarity* S_{u-a}, relates the user vector $\mathbf{v}_{u_t}$ to every check-in vector of this user ($\mathbf{v}_{c,u}$). This similarity indicates the time-wise user-preferred venue categories. The second one, *user-location similarity* S_{u-l}, relates the user vector $\mathbf{v}_{u_t}$ to every location vector $\mathbf{v}_l$. This similarity indicates the user-preferred locations in each venue category.

During the recommendation stage, given a timeslot t, we first select the C most favored venue categories of the user based on the *user-activity similarity* S_{u-a} and list these categories in descending order $f_1, \ldots, f_C$. Now, we focus on unique locations within selected categories. We mark the aforementioned *user-location similarity* S_{u-l} as $S_{u-l,original}$. On the basis of it, for each location, we calculate its distance *dist* to the user coordinate centroid in this timeslot ($coordinate_{u_t}$). Considering the category preference order and location-to-centroid distances, we introduce two exponential decay factors, *category decay CD* and *spatial decay SD*, modeling the likelihood of a user straying from their usual categorical and spatial patterns.

$$CD = a_1 \times \exp(-a_2 \times f_c), \tag{8}$$

$$SD = b_1 \times \exp(-b_2 \times dist), \tag{9}$$

where $a_1, a_2, b_1, b_2 \in \mathbb{R}$, $f_c \in \{0, 1, \ldots, C-1\}$. Inspired by an intuitive heuristic which works widely in practice [14, 20], we multiply the original user-location similarity with these two decay factors to calculate the final user-location similarity ($S_{u-l,final}$) as

$$S_{u-l,final} = S_{u-l,original} \times CD \times SD. \tag{10}$$

Afterwards, we sort all locations belonging to these C categories in descending order of $S_{u-l,final}$ and make recommendations from the top. We refer to this algorithm as the *Spatial-Temporal Embedding Similarity algorithm* (STES) and use the acronym STES in the rest of the article.

4 EXPERIMENTS AND EVALUATION

In this section, we begin by introducing the experimental dataset and data pre-processing details; then we elaborate on the various experiments and evaluate the results.

4.1 Dataset

As described in previous sections, the dataset is required to contain check-in time, location, and the functional role of the visited venue. A robust and popular method to define venue functional roles is to leverage the Foursquare hierarchy of venue categories. The Foursquare venue category tree has four hierarchical levels. In our work, we utilize the second-level categories containing 422 classes such as *American Restaurant*, *Bar*, and *Metro Station*. This is motivated by two reasons. First, there exist only 10 top-level labels, which are too coarsely divided to differentiate places. Secondly, third- and fourth-level categories are too specific to cover all of the venues. In contrast, second-level categories achieve the best sparsity-specificity tradeoff.

Among datasets containing Foursquare check-ins, we select a publicly available one from [5] for three reasons.

— **Space and time span.** This dataset contains globally collected check-ins across 11 months from Feb 25, 2010 to Jan 20, 2011, providing over 12 million Foursquare check-in records with the global spread that we require for our experiment about model generalization (RQ5).

— **Sufficient information.** In this dataset, each raw check-in entry involves user ID, location coordinates, time, venue ID, and source URL. Although the venue category is not originally included, it can be crawled via Foursquare's venue search API[3] using the source URL.
— **Comparability.** This dataset has been utilized in a variety of relevant works, including location recommendation [16, 17, 64], urban activity pattern understanding [10, 11], location-based services with privacy awareness [46, 62], and so forth.

Another two Foursquare datasets from [28] and [76] are also leveraged in relevant pieces of research. However, the former only contains check-ins in Singapore while the latter does not include any venue category. Therefore, we exclude them from our study.

A common issue in check-in data streams are repeated check-ins at the same venue in an artificially short time window during which users remain in an unchanged location and activity [26]. This appears reasonable as check-ins are often posted in a casual way in which people share real-time affairs and moods. This effect is especially common in recreational and culinary activities. For instance, a user may check in for several times during one meal with friends, each time posting about a newly served dish or commenting on the food. To model activity sequences most reliably, we delete such repeated check-ins from an individual staying in an unchanged activity and retain only the first check-in at this location. To reduce noise, we further remove both users and locations with less than 10 posts. After this pre-processing, for the example of New York City (NYC), our dataset contains 225,782 check-ins by 6,442 users at 7,453 locations.

To define urban "neighborhoods" for our area-centric tasks *urban functional zone study* and *crime prediction*, we utilize official 2010 Census Block Group (CBG) polygons,[4] matching the time period of the check-in data collection. A CBG may contain several Census Blocks (CBs), which are the smallest geographic areas that the U.S. Census Bureau uses to collect and tabulate census data. These polygons represent the most natural segmentation of a city, given that their boundaries are defined by physical streets, railroad tracks, bodies of water as well as invisible town limits, property lines, and imaginary extensions of streets. In NYC, there are 6,493 CBGs, 1,720 of which are populated by our denoised check-ins.

4.2 Qualitative Analysis of Embedding Vectors

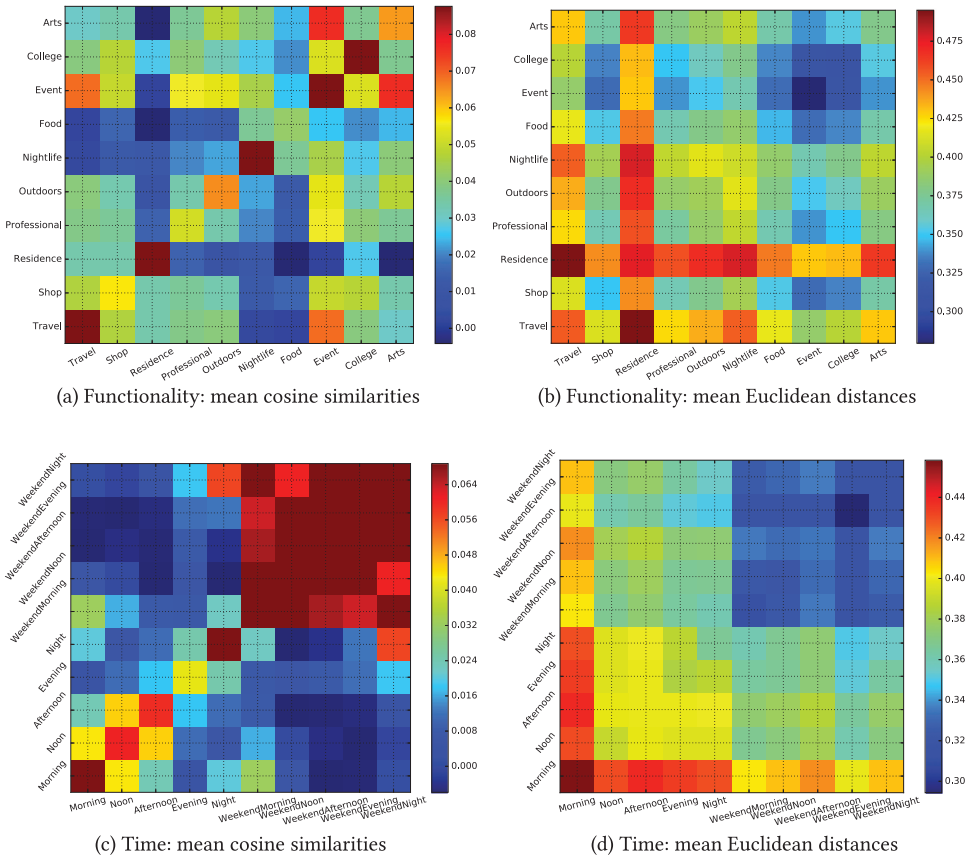
Our embedding vectors are designed to retain key characteristics of user activities and urban venues so that users and places are differentiable. We set the latent embedding dimension to 200 and obtain embedding vectors for all of the *feature words* and *location words* in NYC.

To get a qualitative impression of the resulting embeddings, we first examine their overall cosine similarities and Euclidean distances. The cosine metric evaluates the similarity by normalizing vectors and measuring the in-between angle, and the Euclidean distance demonstrates the magnitude of difference between two vectors.

We explore *feature word* embedding vectors from both functionality and time perspectives. We first calculate the cosine similarity and Euclidean distance for each pair of embedding vectors, then we compute the mean similarity and distance values among the timeslots and top-level venue categories, respectively. We demonstrate the results in the form of heatmaps in Figure 5, where four significant tendencies can be observed.

In Figure 5(a), we can see that intra-category embedding vectors show the highest mean cosine similarities. The single exception *Arts* lists *Arts-Arts* similarity as second to the *Arts-Event* pair. Considering that *Event* mainly involves places for sport, musical, and arts activities, this result

[3] https://developer.foursquare.com/docs/venues/venues.
[4] http://data.beta.nyc/dataset/2010-census-block-groups-polygons.



(a) Functionality: mean cosine similarities

(b) Functionality: mean Euclidean distances

(c) Time: mean cosine similarities

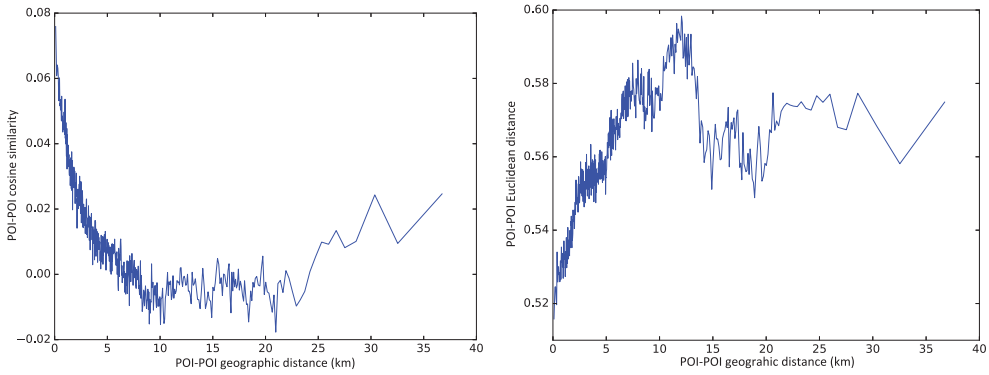
(d) Time: mean Euclidean distances

Fig. 5. Heatmaps of mean cosine similarities and Euclidean distances for feature word embeddings. We normalize the Euclidean distances by the distance range.

appears reasonable. Categories including similar or overlapping venues also have good inter-class similarities, such as *Food-Nightlife*, *College-Event*, and *Outdoors-Event*.

Turning to Figure 5(b), the mean intra-category Euclidean distances of *College* and *Event* are smaller than inter-category distances, which indicates that these two categories have the most compact embedding vector clouds. *Arts*, *Food*, *Nightlife*, *Outdoors*, *Professional*, and *Shop* also have moderate intra- and inter-class Euclidean distances. *Residence* and *Travel* have the largest intra- and inter-category Euclidean distances, which implies that embedding vectors belonging to these two categories are widely distributed in the embedding space. The overall scenario is akin to a geographic area in reality: *Travel* and *Residence* spots are usually spread over the city, while places with highly specific functions such as *College* are more concentrated in a small district. Venues related to *Food*, *Nightlife*, *Shop*, and *Professional* (mainly including places for work, public services, and medical treatments) are widely spread but usually there also exist specific districts such as a central business district (CBD), mainly reserved for these activities.

Figure 5(c) shows that check-ins on weekends are more similar to each other while weekday-weekend similarities are less significant. This indicates people's different activity patterns on weekdays and weekends. For each timeslot on weekdays, the highest mean cosine similarity comes from the intra-timeslot case. We can also see that the difference between working hour activities



(a) Cosine similarity of location pairs (y-axis) with increasing geographic distance (x-axis)

(b) Euclidean distance of location pairs (y-axis) with increasing geographic distance (x-axis)

Fig. 6. Cosine similarity and Euclidean distance of location pairs as a function of geographic distance (in *km*). As before, we normalize Euclidean distances by distance range.

and after-work life is delivered by the similarities within daytime (*Morning*, *Noon*, and *Afternoon*) and nighttime (*Evening* and *Night*) embedding vectors. It also demonstrates the continuity of time under our model. For instance, *Morning* vectors are similar to *Noon* vectors, and *Noon* vectors are similar to *Afternoon* vectors. However, *Morning*-*Afternoon* similarity is much less significant.

Figure 5(d) shows that vector clusters for weekend timeslots are more compact compared to weekday ones. As intra-timeslot vectors involve all of the activities taking place in that period of time, this distance heatmap underlines the fact that in contrast to the weekend life which mainly involves leisure and recreation, diversity is much greater on weekdays where professional and educational activities are included as well.

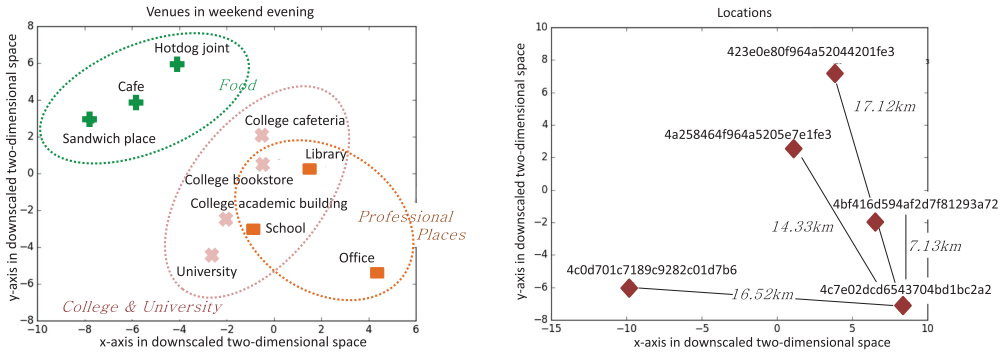
Now we examine the *location word* embedding vectors. As for *feature word* vectors, we first calculate the cosine similarity and Euclidean distance for each pair of locations. We also calculate their geographic distances in the physical world. We sort the resulting triples (geographic distance, cosine similarity, Euclidean distance) according to increasing geographic distances.

To display the results in a legible and informative manner, we divide raw sequences into short segments of length 500. We calculate their mean distances and similarities per segment and plot the results in Figure 6.

It can be seen that within approximately $10km$, as the geographic distance between locations increases, there is a decreasing tendency of the cosine similarity and a reverse trend of the Euclidean distance between their embeddings. Beyond the $10km$ point, both curves demonstrate fluctuation and rebounding trends. Such variations reflect the setup of real urban spaces where districts of similar functionalities are distributed periodically along geographic distances.

To further verify this relationship, we measure the correlation between geographic distance and cosine similarity/Euclidean distance using Pearson correlation coefficients and Spearman rank-order coefficients, respectively. Both coefficients range between -1 and 1 with 0 implying no correlation.

We begin by globally measuring correlations for all POI-POI pairs and note a moderate-to-strong connection between geographic distance and cosine similarity (Pearson: -0.576, Spearman: -0.763). The correlation between geographic distance and vector space Euclidean distance is very similar (Pearson: 0.555, Spearman: 0.788).



(a) *feature word* embeddings (b) *location word* embeddings

Fig. 7. *Feature word* and *location word* example vectors in their semantic spaces.

As observed earlier, in Figure 6, the connection between physical and vector space distances seems more pronounced within a range of 10km. To account for this fact, we further measure correlation coefficients for those POI-POI pairs whose geographic distance does not exceed 10km. In this setting, we note a near perfect correlation between physical distance and cosine similarity (Pearson: −0.851, Spearman: −0.934) as well as Euclidean distance (Pearson: 0.933, Spearman: 0.953).

In all of the above cases, the accompanying p-values are much smaller than 0.001, suggesting that the observations are stable.

To more tangibly and intuitively showcase the trained embedding vectors in their semantic spaces, we project and plot the original 200-dimensional vectors into a two-dimensional space. We experimented with various algorithms including isometric mapping (Isomap), multi-dimensional scaling (MDS), t-distributed stochastic neighbor embedding (t-SNE), and finally arrived at MDS as implemented in Python's scikit-learn package[5] as it produces the most informative visualizations. Examples of these projected vectors are depicted in Figure 7.

Figure 7(a) shows some venue embeddings during weekend evenings. *Food* venues, *College& University* venues, and *Professional* venues are clustered into three groups with the latter two sharing a considerable overlap. This demonstrates the embedding's ability to retain functional correlations that might not have been expressed by discrete venue category labels. More specifically, *School* and *Library* are much closer to *College&University* places than to other professional places such as *Offices* that do not serve an educational purpose.

Location word embeddings reflect geographic proximity among venues. In Figure 7(b), we take location *4c7e02dcd6543704bd1bc2a2* as an example. It can be seen that vector-to-vector Euclidean distances qualitatively correspond to location-to-location geographic distances.

This section aims to qualitatively answer **RQ1**. *Feature word* vectors and *location word* vectors are embedded with functional, temporal, and geographic similarities. Annotated with such embedding vectors, places and users are represented in a way that reflects location visiting patterns as well as people's activity preferences. In the following, we will quantitatively show how the embedding vectors can be used for specific problems.

4.3 Location Recommendation

The goal of location recommendation is to predict a list of top-k locations that a specific user may visit given a reference timeslot. For each user, we choose his/her first 80% check-ins as training data and the remaining 20% as test data.

[5]http://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html.

Table 3. Location Recommendation Performance Evaluation of STES Algorithm and Its Variants

	STES	Variant 1	Variant 2	Variant 3	Variant 4	Variant 5
acc@1	**0.105**	0.041	0.036	0.092	0.055	0.057
acc@5	**0.176**	0.077	0.063	0.154	0.103	0.109
acc@10	**0.199**	0.089	0.078	0.189	0.126	0.131

Before comparing with other state-of-the-art recommendation algorithms, we first examine several variants of our proposed method in Section 3.

Variant 1: Check-ins are represented only by their *feature words*.

Variant 2: Check-ins are represented only by their *location words*.

Variant 3: Both *feature word* and *location word* embeddings are applied to represent a check-in. However, the spatial decay is calculated based on the distance from the most recent check-in instead of the historic user coordinate centroid.

Variant 4: Only *feature word* embeddings are applied to represent a check-in but we adjust the embedding training process. We utilize the *location word* at the output layer to tune model weights.

Variant 5: Similar to Variant 4, we only utilize the *feature word* embeddings but adjust the training process by adding a subsequent second training round. *Feature words* are leveraged as the output and the input layer of the first and the second training round, respectively. *Location words* are utilized as the output layer of the second training round.

The first three variants are mainly related to the recommendation algorithm described in Section 3.3 and the last two variants change the embedding training process elaborated in Section 3.2.

The latent embedding dimension is set to 200. We individually tune the parameters of each variant to reach optimal performance before comparing them with our proposed STES algorithm in terms of top-k accuracy as utilized in [17]. The top-k accuracy for a test check-in is 1 if the ground-truth location is in the top-k recommendations and 0 otherwise. We demote the metric as acc@k and report the average top-1, top-5, and top-10 accuracies over all test check-ins in Table 3.

From Table 3 we can see that the STES algorithm outperforms all its variants. Especially when altogether removing either *feature* or *location words*, we experience harsh performance losses as compared to the overall model. We will now move on to comparing STES with a representative range of state-of-the-art location recommendation algorithms.

STT [17]: This is a topic model based method with consideration of geographic influence and temporal activity patterns. Based on user-specific and time-specific topic distributions, the model selects a check-in topic and recommends a location according to the topic and time-dependent location distributions.

GT-SEER [76]: This algorithm is based on neural embedding techniques. To represent geographic influence, it compares the distance between two places with a threshold, and then explicitly defines neighboring places. It also models the temporal variance into latent location representations during the embedding vector training process.

TA-PLR [33]: Another embedding-based approach. Given check-in sequences, it trains embeddings for location IDs. It also trains latent representation vectors for each time frame and each user. Location recommendation is then based on temporal user-location preference.

Rank-GeoFM [28]: This is a ranking-based factorization method incorporating temporal and geographic influence. Assuming that location preference is relative to the check-in frequency, it fits the users' preference rankings for places to learn the latent factors of users and places. This method is further discussed in [34], in which 12 recommendation schemes are evaluated. Based on four

Table 4. Location Recommendation Performance Evaluated by Precision, Recall, Accuracy, and MAP

	p@1	p@5	p@10	r@1	r@5	r@10	acc@1	acc@5	acc@10	MAP@1	MAP@5	MAP@10
LRT	0.371	0.105	0.054	0.016	0.038	0.052	0.017	0.033	0.059	0.017	0.029	0.042
Rank-GeoFM	0.42	0.152	0.069	0.019	0.046	0.067	0.021	0.051	0.077	0.021	0.045	0.058
GT-SEER	0.462	0.179	0.099	0.021	0.058	0.076	0.047	0.088	0.126	0.047	0.071	0.093
TA-PLR	0.457	0.184	0.096	0.025	0.062	0.087	0.045	0.101	0.143	0.045	0.074	0.091
STT	0.547	0.209	0.125	0.032	0.071	0.09	0.055	0.137	0.174	0.055	0.084	0.098
GEmodel	0.598	0.224	**0.152**	0.057	0.085	0.108	0.083	0.169	**0.201**	0.083	0.124	0.130
STES	**0.606**	**0.227**	0.147	**0.064**	**0.089**	**0.109**	**0.105**	**0.176**	0.199	**0.105**	**0.128**	**0.132**

widely used metrics precision, recall, normalized discounted cumulative gain, and mean average precision, the Rank-GeoFM method consistently performs the best among different datasets and user types.

LRT [8]: This matrix factorization based method measures temporal influence to capture user-location preference and makes recommendations accordingly.

GEmodel [64]: This work introduces a graph-based embedding model. The authors design four bipartite graphs to encode sequential effects, geographical influence, temporal cyclic effects, and semantic effects, respectively, and train embedding vectors to represent user, time, region, and POI for user preference ranking calculation.

In addition to the previously studied top-k *accuracy*, we also compare individual system performance in terms of *precision*, *recall*, and *mean average precision (MAP)*, all of which are common choices in location recommendation evaluation [17, 32, 76]. Similar to accuracy, we denote these metrics at top-k recommendation as p@k, r@k, and MAP@k, respectively. Their definitions are formulated as follows:

$$p@k = \frac{1}{|U|} \sum_{u \in U} \frac{|L_{gt} \cap L_{rec}|}{k}, \tag{11}$$

$$r@k = \frac{1}{|U|} \sum_{u \in U} \frac{|L_{gt} \cap L_{rec}|}{|L_{gt}|}, \tag{12}$$

$$MAP@k = \frac{\sum_{t \in T} AveP(t)}{|T|} \tag{13}$$

in which, U represents the user set, L_{gt} and L_{rec} represent the set of ground-truth locations and the set of corresponding recommended locations for each user in the test data. T represents the total test dataset, and $AveP(t)$ refers to the average precision for each test case.

To find the individually best performance of each method, we tune parameters on the training set using cross validation. The best results for each method are listed in Table 4 and correspond to the following parameters:

STT: Counts of latent regions and topics are both 150.
GT-SEER: Distance threshold is $1km$; 50 unchecked locations, β/α is 0.5; embedding size is 200.
TA-PLR: α is 1; λ is 0.01; embedding size is 200.
Rank-GeoFM: $\alpha, \gamma, \epsilon, C, K$ are 0.15, 0.0001, 0.3, 1, 100; 300 nearest neighbors.
LRT: Latent d is 10; α, β, λ are 2, 2, 1.
GEmodel: Embedding size is 100; 150 samples; the time interval is 30 days.

STES: Embedding size is 200; 15 most preferred venue categories; decay parameters a_1, a_2, b_1, b_2 are 0.4, 0.025, 1, 0.95.

We find that our STES algorithm and the GEmodel produce very similar results, which are both significantly better than those of all other baseline methods. GEmodel marginally beats our algorithm in terms of precision and accuracy at top-10 recommendations while in all other cases our STES model is slightly better. However, GEmodel was specifically designed for location recommendation. Although the embeddings can also be utilized in other domains, it is less flexible and general than our algorithm. In the rest of this article, we will show how our algorithm can be gracefully generalized to other tasks without any adjustments.

We confirm the statistical significance of performance differences between our method and all of the contesting baselines using McNemar's test [37]. The largest mid-p-value is 1.53×10^{-4}.

With respect to **RQ2**, the results demonstrate the effectiveness of our embedding model and the STES algorithm in user/place characterization for location recommendation. As geographic and temporal aspects are considered in these six approaches in different stages, we argue that our improvement mainly comes from the embedding of venues' functional roles. In addition to indicating *where* and *when* someone is, the functional information further explains *why* someone is there at that time, essentially revealing a person's activity preference beyond specific location preference. Consequently, we attain better relative modeling power when a user is in a region which is away from his/her frequently visited area and literal check-ins at unique locations cannot be levied. Moreover, calculating the *mean* of check-in vectors further leverages the continuity and smoothness of the embedding model and thus establishes more latent correlations between users and locations.

4.4 Urban Functional Zone Study

Previous work has shown that dividing a city into different functional zones is a straightforward yet informative way to define urban areas. The central information according to which to partition functional zones are the inhabitants' interactions with urban spaces. Therefore, we conduct research in this aspect to examine model efficiency in describing people's activities and characterizing places. To do so, we exclusively utilize *feature word* embeddings which contain second-level venue categories and check-in timestamps. We train the embedding model on neighborhood level and represent each neighborhood using the mean of all contained check-in vectors. Then, we implement k-means clustering on neighborhoods as suggested by [41] and [77].

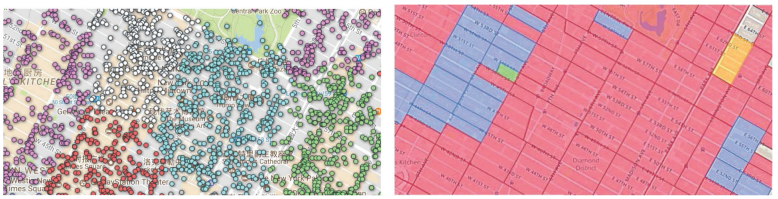
Remember that in location recommendation, we compared our model with a baseline algorithm GEmodel. Similar to our method, GEmodel generates timestamp and venue category embeddings as well. As an additional comparison, we also characterize neighborhoods with the mean of inner check-in time and venue category vectors trained by GEmodel.

Zhu et al. [77] demonstrate an effective neighborhood characterization based on the normalized counts of demographic, temporal, and spatial aspects of visits. Noulas et al. [41] show that the functional zones can be reliably clustered if neighborhoods are represented only by the number of visits at each venue category. Corresponding to these two approaches, we first propose two ground-truth alternatives: (1) l2-normalized counts of *feature words*; (2) l2-normalized counts of venue categories.

To determine the most qualified ground-truth in our work, we examine the cluster assignments derived from the alternatives using Silhouette Index (SI) [47], which measures how compact clusters are by computing the average intra-cluster and inter-cluster distances. The SI ranges between -1 for incorrect clustering and $+1$ for highly dense and well separated clustering. An SI around 0 indicates overlapping clusters.

Table 5. Silhouette Index Measurements

	3	4	5	6	7	8	9	10
Our model	0.315	**0.557**	**0.415**	0.233	0.208	0.252	0.244	0.354
GEmodel	0.297	**0.453**	**0.492**	0.285	0.214	0.259	0.236	0.296
Alternative 1	0.253	**0.487**	**0.443**	0.271	0.209	0.237	0.211	0.323
Alternative 2	0.281	**0.429**	**0.443**	0.231	0.294	0.242	0.198	0.236



(a) POI-based clustering (b) Neighborhood-based clustering

Fig. 8. POI-based clustering utilizes check-in information more directly. However, on a higher geographic abstraction level, this results in a globally less representative clustering. On the other hand, since neighborhoods are Census Block Groups defined by the U.S. Census Bureau, neighborhood-based clustering ensures a natural interpretation of the urban functionality and the daily interaction between people and the their surroundings.

We increase the count of clusters from 3 to 10 and report the Silhouette indices of clusterings in Table 5.

We can see that all compared methods peak in performance at four or five clusters. Compared with GEmodel-based clustering, our embedding model produces more well-defined clusters. Similarly, ground-truth alternative 1 performs better than alternative 2.

We begin by calculating vector representations of neighborhoods and utilize those as the smallest units for clustering. Therefore, our scenario is different from the work described by Cranshaw et al. [7] in which clustering is based on individual POIs. POI-based clustering leverages check-in information in a more direct manner. However, as demonstrated in Figure 8, it is difficult to utilize POI-based clustering results for global study of urban functionality. Another two urban clustering works [74, 77] are excluded from the comparison since both of them require more detailed traces of personal mobility, such as GPS trajectories, which are not available in our dataset. Specifically, they rely on complete travel logs in a period of time during which consecutive leaving and arrival locations and times are recorded. In our scenario, however, check-ins are rather sparse and do not allow for robust computation of such models.

In the following, we focus on our embedding model based clusters and utilize the alternative 1 (l2-normalized counts of *feature words*) as the ground-truth.

Given this ground-truth, a more objective validation of the results can be obtained by comparing the clusters derived from the embedding model with those from the ground-truth, aiming for them to be as similar as possible [77]. A common metric for this scenario is the Rand Index (RI) [45]. This metric penalizes pairwise disagreeing cluster assignments across models. In our work, we employ the Adjusted Rand Index (ARI) [40], which further discounts for expected clustering coherence due to chance.

ARI is bounded in $[-1, 1]$, where 1 corresponds to a perfect match score and random (uniform) assignments lead to a score close to 0. In addition to the clustering based on all check-ins, we further perform clustering based on only daytime (weekday and weekend morning, noon, and

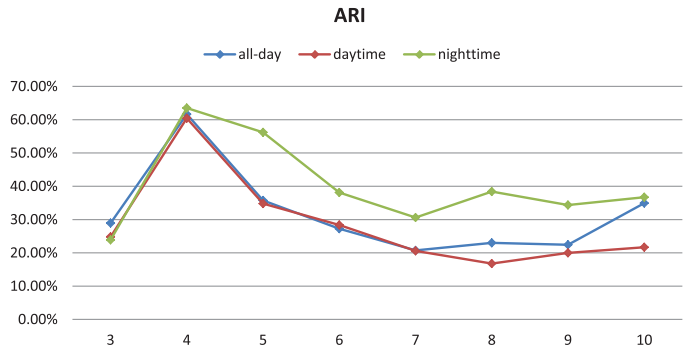


Fig. 9. Adjusted Rand Index (ARI) of embedding model based clustering and ground-truth based clustering. In all of the cases, ARI peaks at around 61% with four clusters.

afternoon) check-ins and only nighttime (weekday and weekend evening and night) check-ins. Figure 9 illustrates the ARIs, showing all three cases peaking at four clusters, incidentally the same point as demonstrated by SI in Table 5.

As a consequence, the following results and evaluations are exclusively based on the four-cluster setting. We interpret results semantically as the clustering is based on concrete functionality. In Figure 10, we plot the composition ratio of venue categories in each cluster based on person-time check-ins. As can be expected from the previously observed high ARI, no matter whether all-day, daytime, or nighttime clustering, clusters based on the embedding model and the ground-truth show high similarities in their compositions and have one-to-one correspondence.

Let us focus on the embedding model based clustering. We find that in most cases, a cluster has a single dominant functionality. For better visualization, Figure 11 depicts some cluster examples. In addition, Table 6 lists exact cluster ratios in the five NYC boroughs. In all cases, the red cluster represents multi-functional zones. Except for daylong daily routines, they contain more professional activities during daytime while experiencing more entertainment at nighttime. Geographically, during the day, red neighborhoods are mainly distributed in Manhattan, Brooklyn, and Queens, which are the most densely populated boroughs involving economical, political, and cultural activities. The orange clusters are primarily for travel and transportation. By examining individual neighborhoods, they contain major expressways, main transportation junctions such as Lexington Avenue Station and the Coney Island Complex. The green cluster represents residential zones. Staten Island, The Bronx, and Queens (especially upper Queens) have higher proportions of the green cluster, which is reasonable as these districts display considerable amounts of residential areas.

By comparison, the blue cluster is assigned different functionalities in different timeslots. Generally speaking, its main roles are related to food and nightlife. During daytime, it is predominately used for professional activities, which include work, education, medical service, spiritual activities, and so forth. Some representative places include New York University Langone Medical Center, Ravenswood Generating Station, and Junior High School 217 Robert A Van Wyck. At nighttime, nightlife is the strongest functional component, followed by food. Manhattan and Brooklyn have the largest proportions of blue neighborhoods, and in fact, most of the blue cells are placed in lower Manhattan and northwestern Brooklyn, where in addition to various restaurants, there also exist many popular bars and pubs like McSorley's Old Ale House, the Bridge Cafe, and the Ear Inn.

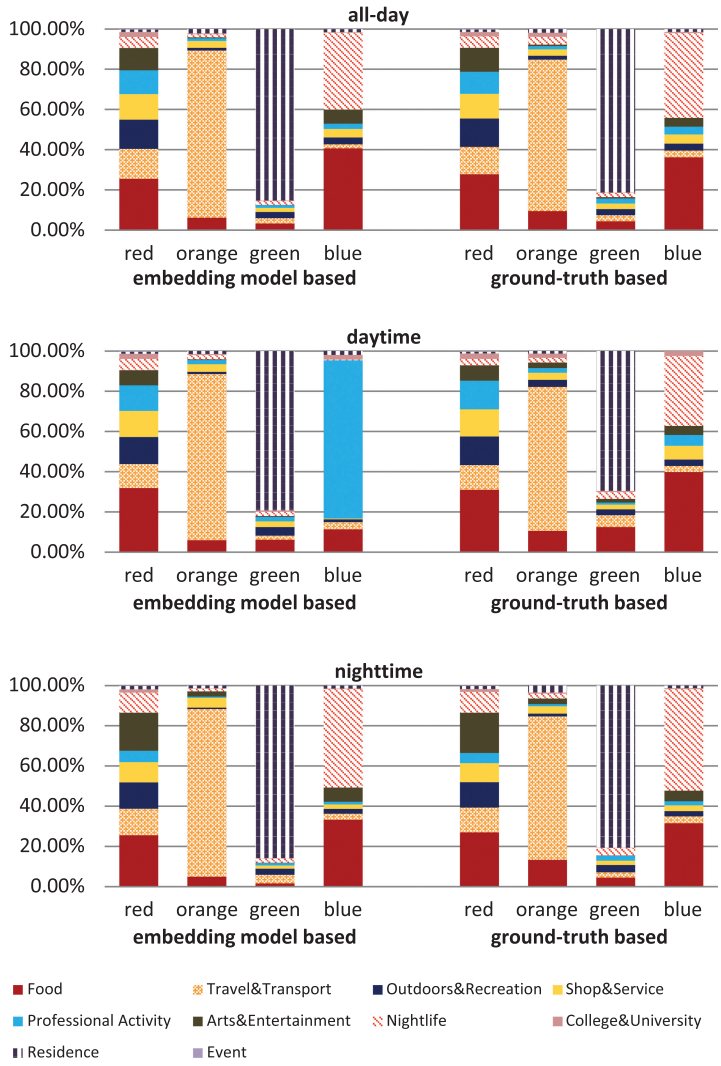
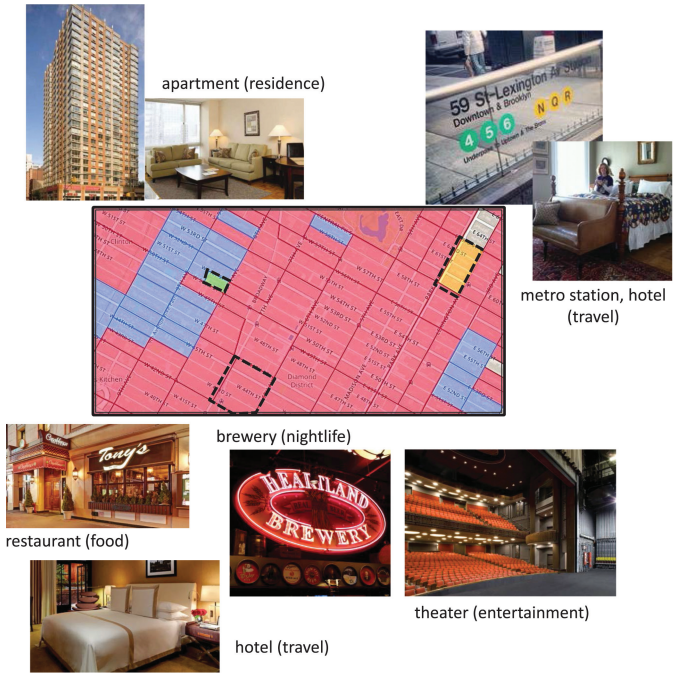
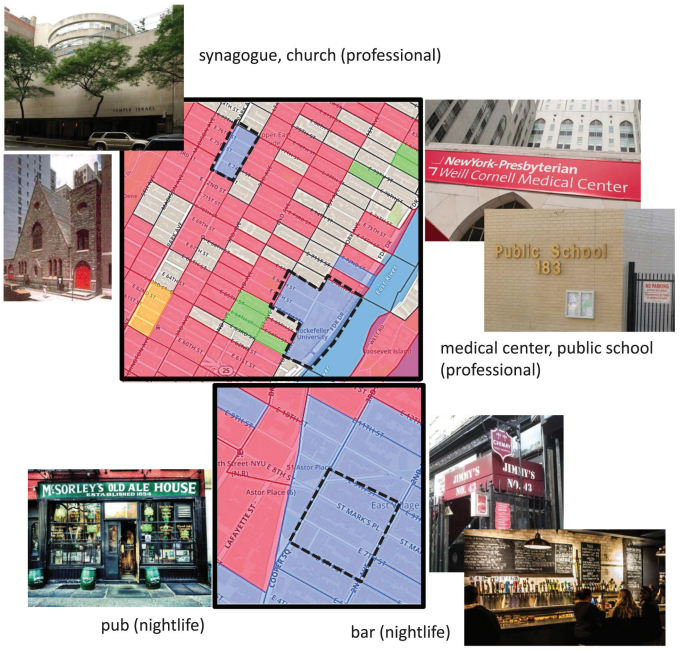


Fig. 10. Functional composition of clusters. The red cluster represents multi-functional zones; the orange cluster is travel&transportation dominant and the green cluster is for residence. The blue cluster is mainly for nightlife and food at nighttime; during daytime, it represents professional area for embedding model based clustering while remaining as food-nightlife place for ground-truth based clustering.

This section discusses **RQ3**. Remember that the embedding model based clustering is purely based on daily check-ins; therefore, it reflects how the city is *actually* used by people. For instance, as the most densely populated area in NYC, Manhattan also boasts some of the city's main transportation hubs and residential buildings. However, orange and green neighborhoods are seldom allocated there. This indicates that the downtown area is generally popular for all types of activities. Moreover, locally popular activities may be time-variant. Figure 12 shows a neighborhood in daytime and nighttime. In daytime, it is mainly used for professional activities, dominated by a clinic, a fire station, and a public school. At nighttime, it is mainly residential, which is plausible as most of the buildings in this area are private homes.



(a) cluster example1

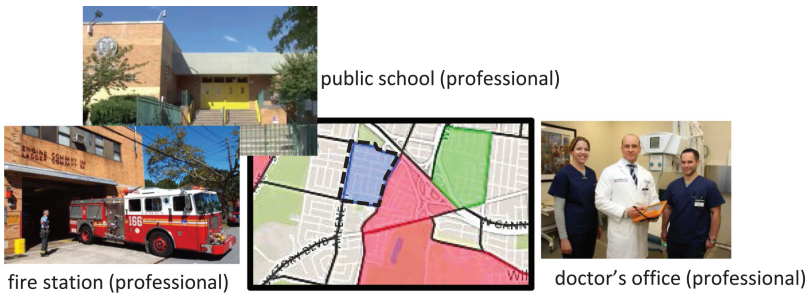


(b) cluster example2

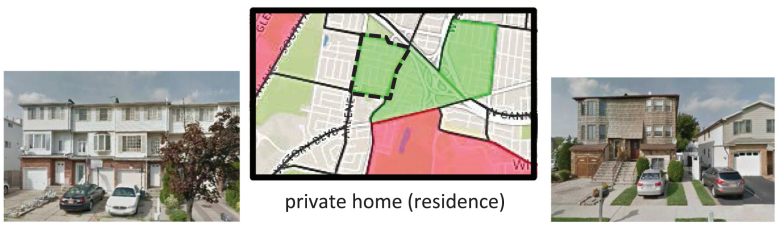
Fig. 11. Cluster examples.

Table 6. Cluster Ratio in NYC Boroughs

		Staten Island	Manhattan	The Bronx	Brooklyn	Queens
all-day	red	58.3%	57.1%	56.6%	48.8%	59.9%
	orange	0%	3%	23.6%	9%	9.1%
	green	30%	5.8%	14.2%	9.9%	12.0%
	blue	11.7%	34.1%	5.6%	32.3%	19.0%
daytime	red	67.8%	**88.7%**	58.5%	**77.3%**	**75.9%**
	orange	0%	3.2%	22.6%	10.2%	9.7%
	green	28.5%	6.2%	14.2%	10.0%	11.4%
	blue	3.7%	1.9%	4.7%	2.5%	3%
nighttime	red	54.4%	54.9%	54.3%	47.9%	55.1%
	orange	0%	2.6%	24.5%	9.1%	9.6%
	green	**33.3%**	6.0%	**16.0%**	10.8%	**14.2%**
	blue	12.3%	**36.5%**	5.2%	**32.2%**	21.1%



(a) neighborhood in the day



(b) neighborhood in the night

Fig. 12. A neighborhood with different functionalities during the day and night. Professional activities are dominant in the day while residence is the main functionality at night.

4.5 Crime Prediction

By providing spatio-temporal embeddings for user and location characterization, our model represents a proxy for the social interactions observed in an urban area. In this section, we will further investigate this capability by addressing a well-known social science problem: crime prediction.

Previous work [57, 61] demonstrates that the occurrence of criminal activities is correlated with place types and time, which are both encoded in our check-in embeddings.

Similar to our urban functional zone study, neighborhoods rather than users are our study subjects in crime prediction. We utilize *feature word* embeddings to characterize neighborhoods. This modification is motivated by the fact that *location words* are only locally descriptive. Therefore, a neighborhood cannot be described without prior training data from the exact location, but such new neighborhood prediction is possible if only modeled with the universally applicable *feature word* vectors, as long as visited venues have the same functional roles. NYC is still the representative city for study and the crime data originates from the NYC Open Data portal.[6]

4.5.1 Crime Rate Prediction. Let us begin by defining the task of future crime rate prediction [9, 60] as predicting the next-month crime rate of a neighborhood. We characterize each neighborhood monthly using the mean of check-in vectors in that month, and we assign crime incidents to neighborhoods according to their location coordinates. A neighborhood is labeled as "Low" crime rate (occurrences=0/month/neighborhood), "Medium" crime rate (0<occurrences<3/month/neighborhood), or "High" crime rate (occurrences>=3/month/neighborhood). Averaging over neighborhoods in each month, 31.3% are of "Low," 37.2% are of "Medium," and 31.5% are of "High" and the variance is 0.047 across months. Averaging across months per location, a neighborhood has a 34.1% chance of "Low," a 39.3% chance of "Medium," and a 26.6% chance of "High" with a variance of 0.087.

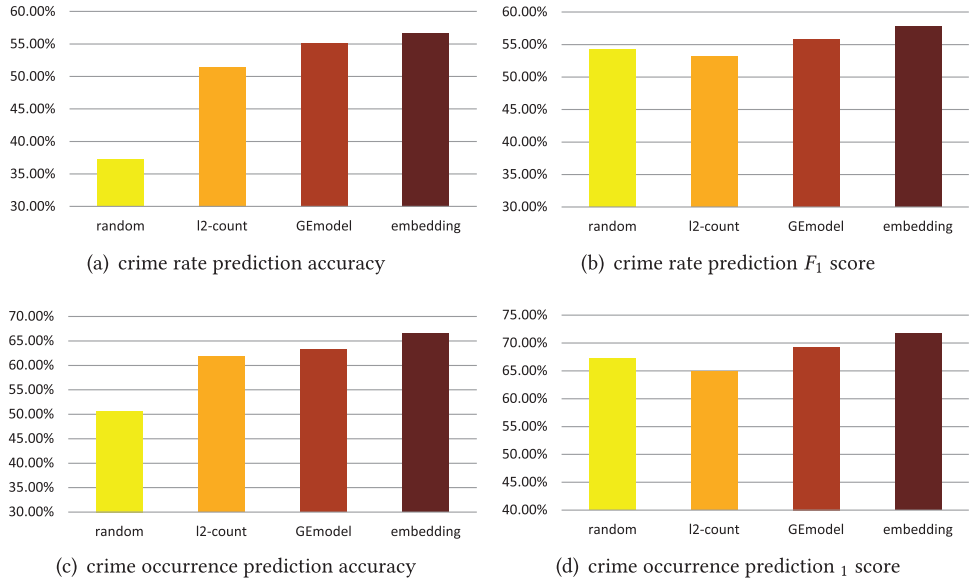
We take all data points from Mar. 2010 to Oct. 2010 (9,983 neighborhoods) as a training set while those in Nov. and Dec. 2010 (2,151 neighborhoods) represent our test set.

We define two performance baselines following our urban functional zone study. The first one is to represent neighborhoods with timestamp and category embeddings trained by GEmodel, and the second one is the l2-normalized monthly counts of *feature words*. Remember that we also reviewed several prediction schemes in the literature part, however, they are not directly applicable in our case due to the lack of tweet texts and demographic statistics such as *education levels*. In addition to these two baselines, we further define a "random" baseline which refers to the results of consistently predicting the most likely label based on the ground-truth of the training data.

We evaluate the performance through accuracy and F_1-score and results are shown in Figure 13(a) and (b). After testing various classification frameworks, we use a random forest classifier. In this case the "random" bar is based on predicting "Medium" crime rate. The figures show that our embedding model produces the best results with significant performance gains over all contestants according to both metrics. We can further note that the F_1-score of random baseline is slightly higher than that of the l2-count baseline method, which is mainly due to the high recall in the three-class random guessing case.

4.5.2 Crime Occurrence Prediction. Aside from predicting overall crime rates, we are interested in understanding local crime events in greater detail. In our crime dataset, a frequently occurring crime type in NYC is *Grand Larceny*. We will now investigate whether we can predict the occurrence of this particular type of crime in a neighborhood within the following month. Experimental settings and classifier choice remain unchanged while the labels are changed into "No Grand Larceny" and "Grand Larceny." When averaging over neighborhoods in each month, the label ratio is 51% for "No Grand Larceny" and 49% for "Grand Larceny" with a variance of 0.027. For each neighborhood, it has on average a 60.9% probability that this crime would occur with a variance of 0.082.

[6] https://data.cityofnewyork.us/Public-Safety/NYPD-7-Major-Felony-Incidents/hyij-8hr7.



(a) crime rate prediction accuracy

(b) crime rate prediction F_1 score

(c) crime occurrence prediction accuracy

(d) crime occurrence prediction $_1$ score

Fig. 13. Average crime prediction accuracies and F_1 scores in NYC.

Figure 13(c) and (d) demonstrate the average prediction accuracies and F_1-scores. Similar to crime rate prediction, we can observe that our model outperforms random guessing as well as both baselines at significance-level.

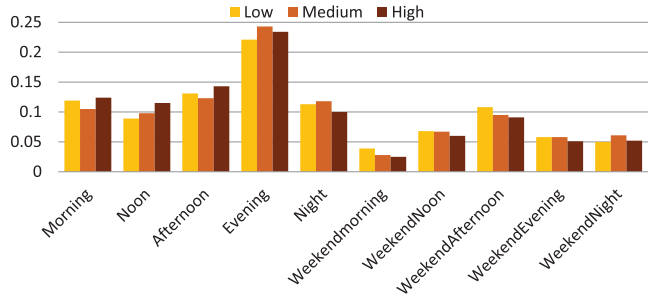
Both crime rate and occurrence prediction results pass the McNemar's significance test with the largest mid-p-value of 1.2062×10^{-7}.

To justify the rationale behind our method and results, we further examine the check-ins in neighborhoods with low/medium/high crime rate and with/without grand larceny, and we plot their check-in time and location category distribution in Figure 14. We can see that neighborhoods with a high crime rate are more checked in on weekdays at entertainment (e.g., casino), shopping, and professional (e.g., business center) places, which also applies to neighborhoods with more grand larceny cases. In response to **RQ4**, this section demonstrates the latent connection between people's daily activities and crime occurrence in an urban area. The prediction results further demonstrate our embedding model's effectiveness in encoding activity information in place representations, furthering the understanding and inference of social science problems.

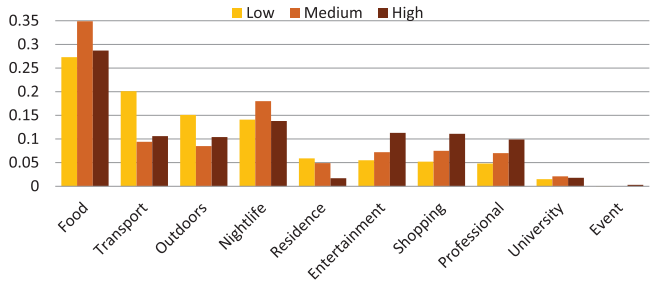
4.6 Model Generalization

The training of large-scale embedding models can be a costly process requiring hours or days of computational resources. To save this time, we investigate whether a pre-trained model can be directly applied in other cities while retaining most of its performance. In this way, our approach differs from existing work in [1], that generalizes user profiles across cities (i.e., the user travels) but still requires local modeling in the new city.

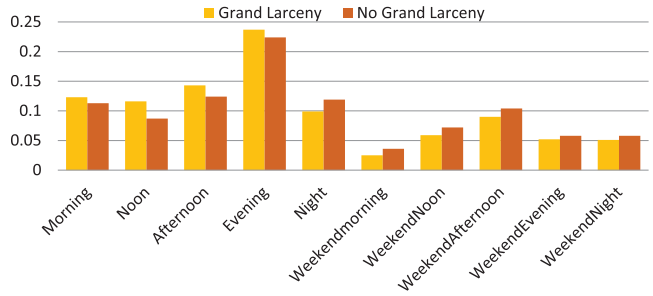
In our generalization experiments, NYC is taken as the reference city where the embedding model is trained. With consideration to data size and the geographic distance to NYC, we select Chicago, Los Angeles, Seattle, San Francisco, London, Amsterdam, Bandung, and Jakarta for the generalization test. As before, we delete repeating check-ins from individuals in artificially short



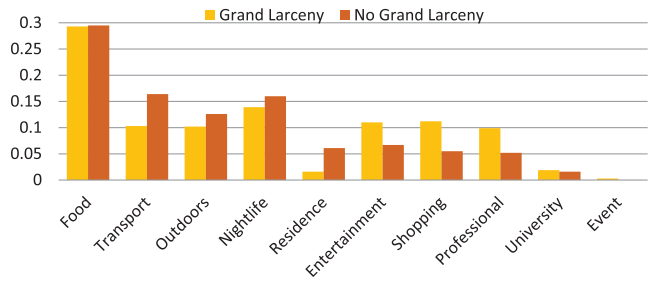
(a) crime rate: time distribution



(b) crime rate: category distribution



(c) crime occurrence: time distribution



(d) crime occurrence: category distribution

Fig. 14. Check-in time and location category distribution in neighborhoods of different labels.

Table 7. Check-In Statistics

City	# of check-ins	# of users	# of locations	Avg. check-ins/user (density)
Chicago (CH)	86,117	2,755	3,678	31.26
Los Angeles (LA)	118,088	4,238	5,609	27.86
Seattle (SE)	44,960	1,523	2,180	29.52
San Francisco (SF)	84,494	3,285	3,605	25.72
London (LO)	45,270	2,182	1,922	20.75
Amsterdam (AM)	49,722	1,855	1,895	26.80
Bandung (BA)	23,581	1,476	996	15.98
Jakarta (JA)	50,875	3,123	1,995	16.29
∗ *NYC*	∗ *225,782*	∗ *6,442*	∗ *7,453*	∗ *35.05*

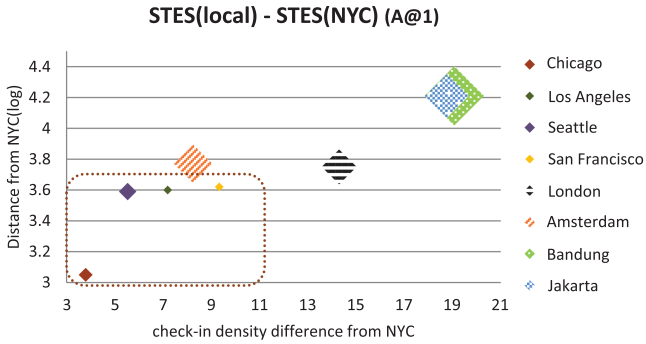
time periods and remove both users and locations with less than 10 posts. Eight cities and their check-in statistics are listed in Table 7. We also list NYC statistics for reference.

4.6.1 Generalization of Location Recommendation. Recall that in our earlier investigation in Section 4.3, we relied on both *feature words* and *location words* for location recommendation. However, since *location words* are locally descriptive, they cannot be easily taken out of their original frame of reference. As a consequence, we only generalize the NYC-based embedding model for *feature words*, but locally train *location words*.

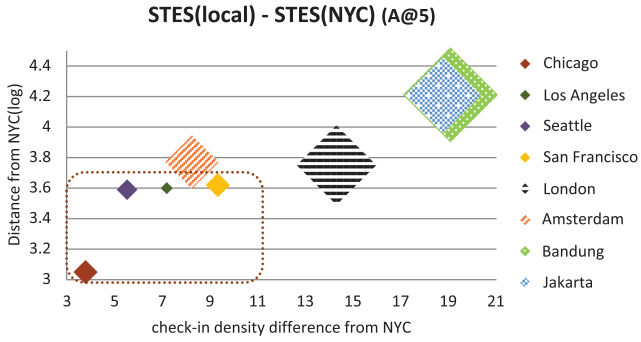
Upon careful examination, none of the baseline methods can be ported across cities. GT-SEER and TA-PLR methods exclusively focus on local POI embedding; Rank-GeoFM and LRT algorithms rely on location vectorization; STT is dependent on local topic model; GEmodel trains embedding vectors based on bipartite graphs but all of the graphs involve local POIs. Therefore, we have to train the baseline methods locally when applying them to different cities.

As before, we measure performance in terms of *precision*, *recall*, *accuracy*, and *MAP* and the results are listed in Table 9. Again, GEmodel and STES(local) have very similar performances, and the half-transferred model STES(NYC) produces close and even better results in some cases. For instance, in Seattle, the NYC-based STES model outperforms both GEmodel and the local STES model according to *recall* at top-1 recommendation. This can happen as a consequence of data sparsity since Seattle only offers 45k local check-ins which provide less information than the 226k transferred ones from NYC.

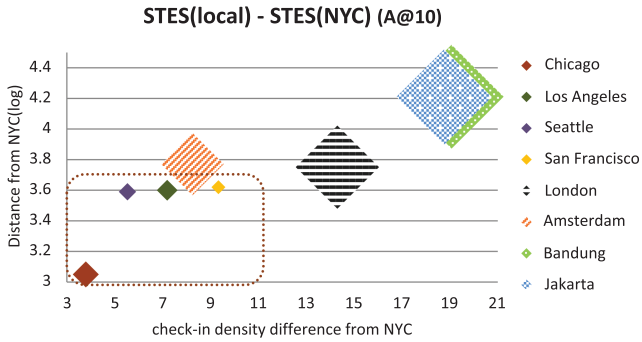
Let us now focus on the STES(NYC) and STES(local) models. Upon closer examination, in all of the eight cities, the gaps between STES(NYC) and STES(local) are generally smaller than 3%. In particular, these two methods almost tie at top-1 recommendation for all U.S. cities; while in the other four non-U.S. cities, the NYC-based STES model produces less satisfactory performance with respect to the local models. To further understand this observation, we plot the accuracy differences between STES(local) and STES(NYC) in top-1, top-5, and top-10 cases in Figure 15. From the figure, we can see that both the difference in local check-in density and the distance from NYC appear to exert an influence on generalization performance. While the former is less significant, the latter plays a key role in model portability as indicated by comparison among Los Angeles, Amsterdam, and San Francisco. We argue that the geographic distance from NYC is a proxy for cultural differences in the way that urban zones are used. Specifically, life style in Southeast Asia is distinct from that in the U.S., which also applies for Europe where the difference seems smaller. Therefore, the NYC-based embedding model is well adapted to other U.S. cities but somewhat less competitive in European and Asian cities.



(a) STES(local) and STES(NYC) gap at A@1



(b) STES(local) and STES(NYC) gap at A@5

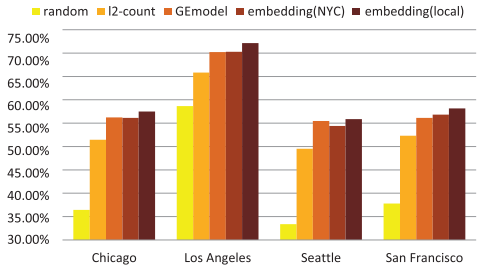


(c) STES(local) and STES(NYC) gap at A@10

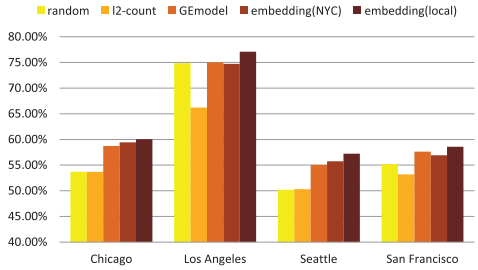
Fig. 15. Accuracy differences (represented by rhombus area sizes) between STES(local) and STES(NYC) for top-1, top-5, and top-10 recommendations. X-axis is the check-in density difference from that in NYC and Y-axis is the log distance in km from the city to NYC. U.S. cities are framed in a brown dash grid. Accuracy differences become larger with increasing distance and check-in density difference.

Table 8. Crime Statistics in U.S. Cities

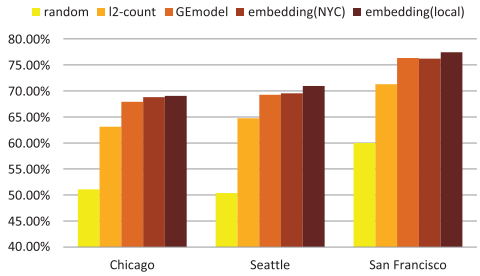
City	Training	Test	Low:(Medium):High	No:Yes
CH	4,668	964	36.7%:32.1%:31.2%	48.9%:51.1%
LA	5,488	1,206	59.8%:40.2%	NA
SE	1,588	324	33.2%:33.5%:33.3%	50.4%:49.6%
SF	2,280	476	31.7%:38.1%:30.2%	40.0%:60.0%



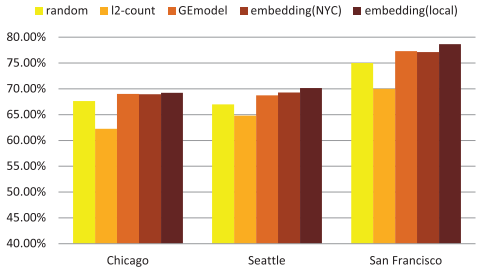
(a) Crime rate prediction accuracy



(b) Crime rate prediction f1 score



(c) Crime occurrence prediction accuracy



(d) Crime occurrence prediction f1 score

Fig. 16. Generalized cross-city crime rate and occurrence prediction in U.S. cities.

4.6.2 Generalization of Crime Prediction. Due to the lack of comparable publicly available crime statistics, generalization experiments are only conducted for U.S. cities. The problem scenario and experimental settings remain unchanged from the description in Section 4.5. Locally collected and processed,[7,8,9,10] crime types vary among cities. Like in NYC, we implement three-grade crime rate prediction with different crime rate thresholds in each city, except for Los Angeles, for which we conduct a two-category experiment since only a very limited amount of crimes were recorded. This is also the reason for the lack of a single frequent crime type in Los Angeles while the other crime datasets report locally frequent crime types. For brevity's sake, we will focus on *Criminal Damage* for Chicago, *Vandalism* for San Francisco, and *Property Damage* for Seattle as they are locally common and classify the neighborhoods more evenly than other crime types. Table 8 lists the training set size, test set size, crime rate ratio, and crime occurrence ratio in each city.

Figure 16 shows the prediction results. In terms of accuracy, local embedding models perform best in all cases. NYC-based STES and GEmodels result in comparable performances, falling behind

[7] https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2.
[8] https://data.seattle.gov/Public-Safety/Crimes-2010/q3s4-jm2b.
[9] http://shq.lasdnews.net/CrimeStats/CAASS/desc.html.
[10] https://data.sfgov.org/Public-Safety/SFPD-Incidents-from-1-January-2003/tmnf-yvry.

Table 9. Location Recommendation Performance in Eight Cities

		p@1	p@5	p@10	r@1	r@5	r@10	acc@1	acc@5	acc@10	MAP@1	MAP@5	MAP@10
	LRT	0.427	0.075	0.053	0.014	0.032	0.049	0.011	0.032	0.044	0.011	0.026	0.032
	Rank-GeoFM	0.501	0.118	0.066	0.023	0.047	0.061	0.019	0.052	0.065	0.019	0.038	0.054
	GT-SEER	0.553	0.121	0.076	0.026	0.053	0.069	0.028	0.071	0.099	0.028	0.062	0.077
CH	TA-PLR	0.672	0.196	0.128	0.037	0.065	0.082	0.079	0.144	0.182	0.079	0.108	0.117
	STT	0.685	0.203	0.132	0.041	0.069	0.085	0.071	0.157	0.205	0.071	0.112	0.12
	GEmodel	0.734	**0.262**	**0.163**	0.055	0.091	**0.117**	0.122	**0.206**	**0.241**	0.122	0.148	**0.166**
	STES(NYC)	0.729	0.238	0.143	0.056	0.088	0.104	0.121	0.197	0.231	0.121	0.145	0.152
	STES(local)	**0.736**	0.257	0.16	**0.061**	**0.094**	0.112	**0.125**	0.205	0.239	**0.125**	**0.151**	0.163
	LRT	0.427	0.076	0.032	0.011	0.028	0.03	0.017	0.049	0.062	0.017	0.035	0.048
	Rank-GeoFM	0.464	0.09	0.051	0.014	0.033	0.036	0.023	0.076	0.107	0.023	0.049	0.057
	GT-SEER	0.498	0.115	0.076	0.021	0.039	0.042	0.036	0.095	0.131	0.036	0.062	0.076
LA	TA-PLR	0.524	0.137	0.096	0.023	0.045	0.051	0.05	0.121	0.149	0.05	0.067	0.084
	STT	0.523	0.149	0.102	0.024	0.048	0.056	0.053	0.123	0.154	0.053	0.071	0.082
	GEmodel	0.594	0.227	**0.155**	0.053	0.089	0.105	0.111	0.186	**0.227**	0.111	0.135	0.138
	STES(NYC)	0.61	0.231	0.15	0.055	0.086	0.099	0.112	0.185	0.215	0.112	0.131	0.137
	STES(local)	**0.617**	**0.232**	0.154	**0.059**	**0.096**	**0.111**	**0.114**	**0.189**	0.221	**0.114**	**0.139**	**0.142**
	LRT	0.258	0.063	0.049	0.011	0.018	0.027	0.019	0.042	0.075	0.019	0.36	0.052
	Rank-GeoFM	0.307	0.072	0.058	0.016	0.024	0.041	0.021	0.069	0.093	0.021	0.047	0.063
	GT-SEER	0.334	0.085	0.063	0.019	0.038	0.046	0.039	0.082	0.113	0.039	0.059	0.076
SF	TA-PLR	0.387	0.129	0.088	0.025	0.046	0.058	0.054	0.112	0.159	0.054	0.079	0.093
	STT	0.396	0.141	0.095	0.032	0.051	0.06	0.062	0.121	0.158	0.062	0.084	0.091
	GEmodel	**0.434**	0.173	0.121	0.052	**0.076**	0.084	0.097	0.153	**0.187**	0.097	**0.118**	**0.125**
	STES(NYC)	0.425	0.172	0.122	0.048	0.070	0.082	0.098	0.154	0.182	0.098	0.112	0.117
	STES(local)	0.432	**0.176**	**0.125**	**0.054**	0.075	**0.089**	**0.103**	**0.162**	0.186	**0.103**	0.117	0.124
	LRT	0.452	0.088	0.057	0.013	0.019	0.048	0.01	0.063	0.079	0.01	0.024	0.046
	Rank-GeoFM	0.478	0.111	0.065	0.018	0.026	0.057	0.028	0.081	0.109	0.028	0.039	0.062
	GT-SEER	0.512	0.143	0.079	0.028	0.037	0.066	0.037	0.101	0.145	0.037	0.056	0.074
SE	TA-PLR	0.566	0.189	0.105	0.032	0.043	0.075	0.05	0.131	0.189	0.05	0.074	0.091
	STT	0.574	0.205	0.112	0.037	0.05	0.081	0.069	0.153	0.196	0.069	0.086	0.097
	GEmodel	0.639	0.247	**0.170**	0.052	0.089	0.105	0.118	0.207	0.246	0.118	0.141	0.153
	STES(NYC)	0.643	0.252	0.167	**0.058**	**0.091**	0.104	0.121	0.205	0.242	0.121	0.143	0.15
	STES(local)	**0.645**	**0.257**	0.169	0.054	**0.091**	**0.106**	**0.124**	**0.215**	**0.252**	**0.124**	**0.147**	**0.156**
	LRT	0.324	0.052	0.035	0.017	0.029	0.034	0.018	0.049	0.083	0.018	0.026	0.037
	Rank-GeoFM	0.385	0.07	0.053	0.024	0.036	0.041	0.027	0.076	0.112	0.027	0.048	0.062
	GT-SEER	0.397	0.089	0.054	0.027	0.042	0.049	0.041	0.094	0.131	0.041	0.065	0.078
LO	TA-PLR	0.428	0.124	0.069	0.038	0.056	0.061	0.073	0.158	0.182	0.073	0.101	0.112
	STT	0.432	0.127	0.075	0.042	0.055	0.064	0.079	0.158	0.189	0.079	0.098	0.109
	GEmodel	0.507	0.162	0.096	0.059	0.077	0.085	0.134	0.198	**0.246**	0.134	0.156	0.162
	STES(NYC)	0.501	0.159	0.097	0.051	0.07	0.079	0.131	0.193	0.221	0.131	0.147	0.153
	STES(local)	**0.514**	**0.169**	**0.103**	**0.062**	**0.081**	**0.094**	**0.141**	**0.216**	0.245	**0.141**	**0.158**	**0.163**

(Continued)

Table 9. Continued

		p@1	p@5	p@10	r@1	r@5	r@10	acc@1	acc@5	acc@10	MAP@1	MAP@5	MAP@10
	LRT	0.523	0.074	0.053	0.016	0.032	0.043	0.029	0.101	0.13	0.029	0.047	0.062
	Rank-GeoFM	0.605	0.119	0.068	0.023	0.047	0.056	0.042	0.123	0.153	0.042	0.074	0.08
	GT-SEER	0.632	0.136	0.085	0.036	0.063	0.079	0.061	0.143	0.171	0.061	0.097	0.105
AM	TA-PLR	0.715	0.208	0.122	0.048	0.093	0.116	0.082	0.217	0.273	0.082	0.146	0.159
	STT	0.702	0.195	0.116	0.048	0.089	0.123	0.101	0.205	0.252	0.101	0.154	0.163
	GEmodel	0.771	0.243	0.132	0.075	0.111	**0.143**	0.175	0.265	0.302	0.175	0.199	0.217
	STES(NYC)	0.766	0.234	0.134	0.077	0.112	0.128	0.178	0.267	0.304	0.178	0.2	0.207
	STES(local)	**0.776**	**0.253**	**0.139**	**0.087**	**0.126**	0.141	**0.189**	**0.283**	**0.322**	**0.189**	**0.213**	**0.225**
	LRT	0.289	0.065	0.048	0.012	0.021	0.027	0.01	0.033	0.052	0.01	0.016	0.029
	Rank-GeoFM	0.325	0.067	0.052	0.017	0.026	0.033	0.013	0.040	0.069	0.013	0.026	0.038
	GT-SEER	0.421	0.086	0.059	0.022	0.034	0.041	0.023	0.062	0.087	0.023	0.035	0.058
JA	TA-PLR	0.488	0.099	0.053	0.027	0.049	0.05	0.032	0.082	0.154	0.032	0.047	0.065
	STT	0.532	0.116	0.063	0.028	0.057	0.068	0.048	0.113	0.179	0.048	0.065	0.084
	GEmodel	0.586	0.168	**0.109**	0.057	0.079	0.094	0.128	0.176	**0.219**	0.128	0.143	**0.167**
	STES(NYC)	0.581	0.162	0.091	0.059	0.082	0.093	0.124	0.173	0.19	0.124	0.137	0.141
	STES(local)	**0.596**	**0.176**	0.108	**0.062**	**0.097**	**0.106**	**0.135**	**0.189**	0.218	**0.135**	**0.151**	0.165
	LRT	0.405	0.053	0.037	0.015	0.028	0.034	0.012	0.028	0.046	0.012	0.02	0.028
	Rank-GeoFM	0.452	0.059	0.044	0.018	0.036	0.047	0.018	0.043	0.061	0.018	0.026	0.037
	GT-SEER	0.493	0.082	0.05	0.021	0.046	0.054	0.024	0.062	0.083	0.024	0.031	0.056
BA	TA-PLR	0.555	0.12	0.078	0.036	0.053	0.062	0.044	0.118	0.168	0.044	0.06	0.071
	STT	0.583	0.165	0.102	0.052	0.067	0.089	0.061	0.134	0.182	0.061	0.082	0.097
	GEmodel	0.649	0.224	0.153	0.067	0.102	**0.129**	0.127	0.186	0.217	0.127	**0.164**	**0.172**
	STES(NYC)	0.641	0.215	0.144	0.069	0.092	0.107	0.121	0.179	0.193	0.121	0.148	0.153
	STES(local)	**0.662**	**0.234**	**0.167**	**0.079**	**0.108**	0.127	**0.138**	**0.207**	**0.223**	**0.138**	0.161	0.17

the local STES model by only less than 2%. In terms of F_1-scores, local embedding models still outperform all the other methods. NYC-based embedding models and GEmodels lead to similar patterns as those in accuracy evaluation. Random guessing in some cases results in a comparable performance to NYC-based embedding models due to its high recall.

These generalization experiments answer **RQ5** by demonstrating only mild transfer errors. Like before, we conduct McNemar's test and ascertain that the performance improvement from baselines to the embedding model are significant at the level of 0.05. This property can remove the need for time-consuming local training while maintaining competitive performance. Additionally, based on the performance differences, we can qualitatively compare similarities between pilot cities and the reference city with respect to their inhabitants' daily activities, life style, and even culture.

5 CONCLUSIONS

In this study, we propose an unsupervised embedding model that learns to represent social network check-ins based on their functional, temporal, and geographic aspects in the form of dense numerical vectors in a semantic space. Item correlations are well captured in terms of activity and location similarities.

We show three model applications: *location recommendation*, *urban functional zone study*, and *crime prediction*. Our embedding model based recommendation algorithm STES outperforms a wide range of state-of-the-art methods according to four performance metrics. Urban functional

zone study shows us intuitive patterns about people's activities in different urban areas. Crime prediction demonstrates the model's effectiveness at capturing location properties and verifies the possibility to infer crime rates or occurrences from residents' daily activities. Furthermore, we confirm that the embedding model has good generality and can be trained in well-represented cities before being applied in other places with only a small generalization error.

In particular, we compare our model with a competitive algorithm GEmodel. Both our method and GEmodel are based on neural network techniques considering temporal, geographic, and functional aspects. In the three application domains, GEmodel produces very similar results to our algorithm, especially in location recommendation. However, GEmodel was designed to be trained with more information such as visitors' reviews which are not available in some cases. In addition, GEmodel needs to be trained locally as their embeddings of regions, timestamps, and words are updated together with the local POI embeddings. Our proposed model does not require any information beyond what is available from common location-based service APIs.

There are several interesting lines of future investigation: (1) While the current model has been designed to be easily trained with minimal restrictions regarding task and data, we are interested in incorporating it in task-specific end-to-end architectures to attain further performance improvements. (2) We are interested to explore whether incorporating explicit geographic coordinates can generate more descriptive location embeddings than merely relying on venue IDs. Such embeddings can, for instance, be trained using location neighborhoods which are determined by their coordinates. (3) On the application side, since the model was designed as a general-purpose descriptor, it has the potential to be employed in a broad range of social tasks such as household income prediction or travel time inference.

ACKNOWLEDGMENTS

We thank Zhiyuan Cheng for providing us with the check-in dataset and Bo Hu for sharing their location recommendation algorithm details. We also thank Jan Dörrie, Zack Zhu, Hangxin Lu, and Hongyu Xiao for their insightful discussions.

REFERENCES

[1] Jie Bao, Yu Zheng, and Mohamed F. Mokbel. 2012. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 199–208.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3, (Jan. 2003), 993–1022.

[3] Chen Cheng, Haiqin Yang, Irwin King, and Michael R. Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12)*, 17–23.

[4] Chen Cheng, Haiqin Yang, Michael R. Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, 2605–2611.

[5] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z. Sui. 2011. Exploring millions of footprints in location sharing services. In *Proceedings of the International Conference on Web and Social Media (ICWSM'11)*, 81–88.

[6] Andrew Church, Martin Frost, and Keith Sullivan. 2000. Transport and social exclusion in London. *Transport Policy* 7, 3 (2000), 195–205.

[7] Justin Cranshaw, Raz Schwartz, Jason I. Hong, and Norman Sadeh. 2012. The livehoods project: Utilizing social media to understand the dynamics of a city.

[8] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 93–100.

[9] Matthew S. Gerber. 2014. Predicting crime using Twitter and kernel density estimation. *Decision Support Systems* 61 (2014), 115–125.

[10] Samiul Hasan and Satish V. Ukkusuri. 2014. Urban activity pattern classification using topic models from online geo-location data. *Transportation Research Part C: Emerging Technologies* 44 (2014), 363–381.

[11] Samiul Hasan, Xianyuan Zhan, and Satish V. Ukkusuri. 2013. Understanding urban human activity and mobility patterns using large-scale location-based data from online social media. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*. ACM, 6.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[13] Tieke He, Hongzhi Yin, Zhenyu Chen, Xiaofang Zhou, Shazia Sadiq, and Bin Luo. 2016. A spatial-temporal topic model for the semantic annotation of POIs in LBSNs. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, 1 (2016), 12.

[14] Francis Heylighen and Johan Bollen. 2002. Hebbian algorithms for a digital library recommendation system. In *Proceedings of the International Conference on Parallel Processing Workshops*. IEEE, 439–446.

[15] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsiouliklis. 2012. Discovering geographical topics in the Twitter stream. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, 769–778.

[16] Bo Hu and Martin Ester. 2013. Spatial topic modeling in online social media for location recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 25–32.

[17] Bo Hu, Mohsen Jamali, and Martin Ester. 2013. Spatio-temporal topic modeling in mobile social media for location recommendation. In *Proceedings of the 2013 IEEE 13th International Conference on Data Mining*. IEEE, 1073–1078.

[18] Rizwan Iqbal, Masrah Azrifah Azmi Murad, Aida Mustapha, Payam Hassany Shariat Panahy, and Nasim Khanahmadliravi. 2013. An experimental study of classification algorithms for crime prediction. *Indian Journal of Science and Technology* 6, 3 (2013), 4219–4225.

[19] Donald E. Knuth. 1985. Dynamic Hhuffman coding. *Journal of Algorithms* 6, 2 (1985), 163–180.

[20] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Communications of the ACM* 53, 4 (2010), 89–97.

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

[22] John Krumm and Dany Rouhana. 2013. Placer: Semantic place labels from diary data. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 163–172.

[23] Takeshi Kurashima, Tomoharu Iwata, Takahide Hoshide, Noriko Takaya, and Ko Fujimura. 2013. Geo topic model: Joint modeling of user's activity area and interests for location recommendation. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*. ACM, 375–384.

[24] Juha K. Laurila, Daniel Gatica-Perez, Imad Aad, Olivier Bornet, Trinh-Minh-Tri Do, Olivier Dousse, Julien Eberle, Markus Miettinen, and others. 2012. The mobile data challenge: Big data for mobile computing research. In *Pervasive Computing*.

[25] Jia Li, Hua Xu, Xingwei He, Junhui Deng, and Xiaomin Sun. 2016. Tweet modeling with LSTM recurrent neural networks for hashtag recommendation. In *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN'16)*. IEEE, 1570–1577.

[26] Wen Li, Carsten Eickhoff, and Arjen P de Vries. 2016. Probabilistic local expert retrieval. In *European Conference on Information Retrieval*. Springer, 227–239.

[27] Wen Li, Pavel Serdyukov, Arjen P. de Vries, Carsten Eickhoff, and Martha Larson. 2011. The where in the tweet. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. ACM, 2473–2476.

[28] Xutao Li, Gao Cong, Xiao-Li Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. 2015. Rank-GeoFM: A ranking based geographical factorization method for point of interest recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 433–442.

[29] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 831–840.

[30] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv:1703.03130*.

[31] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. 2013. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1043–1051.

[32] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

[33] Xin Liu, Yong Liu, and Xiaoli Li. 2016. Exploring the context of locations for personalized location recommendations. In *IJCAI* (pp. 1188–1194).

[34] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of the VLDB Endowment* 10, 10 (2017), 1010–1021.

[35] Paolo Massa and Paolo Avesani. 2007. Trust-aware recommender systems. In *Proceedings of the 2007 ACM Conference on Recommender Systems*. ACM, 17–24.
[36] Wesley Mathew, Ruben Raposo, and Bruno Martins. 2012. Predicting future locations with hidden Markov models. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 911–918.
[37] Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* 12, 2 (1947), 153–157.
[38] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv:1309.4168*.
[39] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.
[40] Glenn W. Milligan and Martha C. Cooper. 1985. An examination of procedures for determining the number of clusters in a dataset. *Psychometrika* 50, 2 (1985), 159–179.
[41] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. 2011. Exploiting semantic annotations for clustering geographic areas and users in location-based social networks. *The Social Mobile Web* 11 (2011), 02.
[42] Makbule Gulcin Ozsoy. 2016. From word embeddings to item recommendation. *arXiv:1601.01356*.
[43] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 701–710.
[44] Van-Thuy Phi, Liu Chen, and Yu Hirate. 2016. Distributed representation-based recommender systems in E-commerce. In *DEIM* Forum.
[45] William M. Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 66, 336 (1971), 846–850.
[46] Daniele Riboni and Claudio Bettini. 2013. A platform for privacy-preserving geo-social recommendation of points of interest. In *Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management (MDM'13)*, Vol. 1. IEEE, 347–349.
[47] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20 (1987), 53–65.
[48] Ruslan Salakhutdinov and Andriy Mnih. 2011. Probabilistic matrix factorization. In *NIPS*, Vol. 20. 1–8.
[49] Siddharth Sarda, Carsten Eickhoff, and Thomas Hofmann. 2016. Semantic place descriptors for classification and map discovery. *arXiv:1601.05952*.
[50] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. 2011. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1046–1054.
[51] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 909–912.
[52] Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of ACL*.
[53] Duyu Tang, Bing Qin, Ting Liu, and Yuekui Yang. 2015. User modeling with neural network for review rating prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 1340–1346.
[54] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for Twitter sentiment classification. In *ACL (1)*. 1555–1565.
[55] Mozhgan Tavakolifard and Kevin C. Almeroth. 2012. Social computing: An intersection of recommender systems, trust/reputation systems, and social networks. *IEEE Network* 26, 4 (2012), 53–58.
[56] Guangjin Tian, Jianguo Wu, and Zhifeng Yang. 2010. Spatial pattern of urban functions in the Beijing metropolitan region. *Habitat International* 34, 2 (2010), 249–255.
[57] Peter J. Van Koppen and Robert W. J. Jansen. 1999. The time to rob: Variations in time of number of commercial robberies. *Journal of Research in Crime and Delinquency* 36, 1 (1999), 7–29.
[58] Weiqing Wang, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, and Xiaofang Zhou. 2015. Geo-sage: A geographical sparse additive generative model for spatial item recommendation. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1255–1264.
[59] Weiqing Wang, Hongzhi Yin, Shazia Sadiq, Ling Chen, Min Xie, and Xiaofang Zhou. 2016. Spore: A sequential personalized spatial item recommender system. In *Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering (ICDE'16)*. IEEE, 954–965.
[60] Xiaofeng Wang, Matthew S. Gerber, and Donald E. Brown. 2012. Automatic crime prediction using events extracted from Twitter posts. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. Springer, 231–238.

[61] David Weisburd, Shawn Bushway, Cynthia Lum, and Sue-Ming Yang. 2004. Trajectories of crime at places: A longitudinal study of street segments in the city of Seattle. *Criminology* 42, 2 (2004), 283–322.
[62] Marius Wernke, Frank Dürr, and Kurt Rothermel. 2012. PShare: Position sharing for location privacy based on multi-secret sharing. In *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications (PerCom'12)*. IEEE, 153–161.
[63] Sanjaya Wijeratne, Lakshika Balasuriya, Derek Doran, and Amit Sheth. 2016. Word embeddings to enhance Twitter gang member profile identification. In *Proceedings of the IJCAI Workshop on Semantic Machine Learning (SML'16). New York City, NY: CEUR-WS*, Vol. 7.
[64] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. 2016. Learning graph-based POI embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 15–24.
[65] Xingxing Xing, Wenhao Huang, Guojie Song, and Kunqing Xie. 2014. Traffic zone division using mobile billing data. In *Proceedings of the 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'14)*. IEEE, 692–697.
[66] Wang Yajun and others. 2006. The functionality analysis to urban green space system structural configuration plan [J]. *Journal of Shandong Forestry Science and Technology* 6 (2006), 033.
[67] Mao Ye, Dong Shou, Wang-Chien Lee, Peifeng Yin, and Krzysztof Janowicz. 2011. On the semantic annotation of places in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 520–528.
[68] Mao Ye, Peifeng Yin, and Wang-Chien Lee. 2010. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 458–461.
[69] Hongzhi Yin, Bin Cui, Yizhou Sun, Zhiting Hu, and Ling Chen. 2014. Lcars: A spatial item recommender system. *ACM Transactions on Information Systems (TOIS)* 32, 3 (2014), 11.
[70] Hongzhi Yin, Bin Cui, Xiaofang Zhou, Weiqing Wang, Zi Huang, and Shazia Sadiq. 2016. Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. *ACM Transactions on Information Systems (TOIS)* 35, 2 (2016), 11.
[71] Hongzhi Yin, Weiqing Wang, Hao Wang, Ling Chen, and Xiaofang Zhou. 2017. Spatial-aware hierarchical collaborative deep learning for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2537–2551.
[72] Hongzhi Yin, Xiaofang Zhou, Bin Cui, Hao Wang, Kai Zheng, and Quoc Viet Hung Nguyen. 2016. Adapting to user interest drift for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering* 28, 10 (2016), 2566–2581.
[73] Hongzhi Yin, Xiaofang Zhou, Yingxia Shao, Hao Wang, and Shazia Sadiq. 2015. Joint modeling of user check-in behaviors for point-of-interest recommendation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 1631–1640.
[74] Nicholas Jing Yuan, Yu Zheng, Xing Xie, Yingzi Wang, Kai Zheng, and Hui Xiong. 2015. Discovering urban functional zones using latent activity trajectories. *IEEE Transactions on Knowledge and Data Engineering* 27, 3 (2015), 712–725.
[75] Jia-Dong Zhang, Chi-Yin Chow, and Yanhua Li. 2014. LORE: Exploiting sequential influence for location recommendations. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 103–112.
[76] Shenglin Zhao, Tong Zhao, Irwin King, and Michael R. Lyu. 2016. GT-SEER: Geo-temporal sequential embedding rank for point-of-interest recommendation. *arXiv:1606.05859*.
[77] Zack Zhu, Jing Yang, Chen Zhong, Julia Seiter, and Gerhard Tröster. 2015. Learning functional compositions of urban spaces with crowd-augmented travel survey data. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 22.

Received July 2017; revised January 2018; accepted January 2018