Embedding Hierarchical Structures for Venue Category Representation

MENG CHEN, School of Software, Shandong University, China
LEI ZHU, School of Information Science and Engineering, Shandong Normal University, China
RONGHUI XU, School of Software, Shandong University, China
YANG LIU, Department of Physics and Computer Science, Wilfrid Laurier University, Canada
XIAOHUI YU, School of Information Technology, York University, Canada
YILONG YIN, School of Software, Shandong University, China

Venue categories used in location-based social networks often exhibit a hierarchical structure, together with the category sequences derived from users' check-ins. The two data modalities provide a wealth of information for us to capture the semantic relationships between those categories. To understand the venue semantics, existing methods usually embed venue categories into low-dimensional spaces by modeling the linear context (i.e., the positional neighbors of the given category) in check-in sequences. However, the hierarchical structure of venue categories, which inherently encodes the relationships between categories, is largely untapped. In this article, we propose a venue **C**ategory **E**mbedding **M**odel named **Hier-CEM**, which generates a latent representation for each venue category by embedding the **Hier**archical structure of categories and utilizing multiple types of context. Specifically, we investigate two kinds of hierarchical context based on any given venue category hierarchy and show how to model them together with the linear context collaboratively. We apply Hier-CEM to three tasks on two real check-in datasets collected from Foursquare. Experimental results show that Hier-CEM is better at capturing both semantic and sequential information inherent in venues than state-of-the-art embedding methods.

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

Additional Key Words and Phrases: Venue category representation, hierarchical category structure, multiple context types, check-in data

ACM Reference format:
Meng Chen, Lei Zhu, Ronghui Xu, Yang Liu, Xiaohui Yu, and Yilong Yin. 2021. Embedding Hierarchical Structures for Venue Category Representation. *ACM Trans. Inf. Syst.* 40, 3, Article 57 (November 2021), 29 pages.
https://doi.org/10.1145/3478285

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61906107, the Natural Science Foundation of Shandong Province of China under Grant No. ZR2019BF010, and the Young Scholars Program of Shandong University.
Authors' addresses: M. Chen, R. Xu, and Y. Yin (corresponding author), School of Software, Shandong University, Jinan, China; emails: mchen@sdu.edu.cn, xrhics@163.com, ylyin@sdu.edu.cn; L. Zhu, School of Information Science and Engineering, Shandong Normal University, Jinan, China; email: leizhu0608@gmail.com; Y. Liu, Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo, Canada; email: yangliu@wlu.ca; X. Yu (corresponding author), School of Information Technology, York University, Toronto, Canada; email: xhyu@yorku.ca.

1046-8188/2021/11-ART57 $15.00
https://doi.org/10.1145/3478285

1 INTRODUCTION

Location-based social network (LBSN) platforms such as Foursquare have generated a tremendous amount of human mobility data, usually in the form of check-ins. A typical check-in contains *user ID*, *venue ID*, *venue category*, and *time*, where venue categories (e.g., *Art Museum* and *Japanese Restaurant*) act as a proxy for functions that a particular venue of a given category affords, and check-ins of a user over time form the check-in sequence. Availability of such check-in data has spurred research in capturing venue semantics, which is key to reasoning about **Point of Interest (POI)** similarity, geographic information retrieval, recommender systems, ontology engineering, and so forth [4, 10, 13, 29, 31, 36, 37, 41, 44, 51].

To understand venue semantics, existing studies [5, 6, 19, 30, 40, 42, 46, 47] usually model check-in sequences to learn representations of venue IDs based on representation learning. They usually adopt the framework of word embedding models (e.g., CBOW/SkipGram [26]) and follow the distributional hypothesis that venues occurring in similar contexts tend to have similar semantics and should appear close to each other in the latent embedding space. Mostly the "context" is defined as the venues that immediately precede and follow the given venue within a given range. However, in contrast to text data where many words appear with a high frequency, venue visit in check-in data is extremely sparse. For example, each venue is visited nine times on average during a period of 18 months in Foursquare [39]. The pattern information conveyed by sequences of sparse venue check-ins is thus highly limited. It proves very difficult, if not impossible, to derive effective representations of venue IDs using such contexts alone.

In this work, we aim to learn a latent representation for each *venue category*, instead of a venue ID, as venue categories can be understood as the summarization of venue semantics, and they are much "denser" than venue IDs in terms of occurrences in check-in data. For example, Foursquare contains about 900 categories in total (in contrast to millions of venue IDs), and each category is visited thousands of times on average during 18 months. This makes it more promising to learn an effective venue category representation than venue ID representation. Further, tackling the new category representation learning task, one may exploit the following unique characteristics of check-in data: (1) a user's check-ins over time naturally form sequences of venue categories (e.g., *Japanese Restaurant* $\rightarrow$ *Museum* $\rightarrow \cdots$), which makes it possible to capture the relations between the venue category and its linear context to learn category embeddings, where the *linear context* is the positional neighbors of the given category in a venue category sequence; and (2) venue categories are also organized with a hierarchical structure in LBSNs, in which they are not independent but hierarchically correlated. For instance, Foursquare organizes venue categories using a tree structure. The category hierarchy is able to assist in capturing venue semantics better.

Nonetheless, the task of learning representations of venue categories based on two data modalities, namely, venue category sequences and venue category hierarchy, presents some unique challenges, and existing methods do not directly apply. (1) **Exploiting hierarchical relationships**: In the category hierarchy, categories on a path (e.g., *Food-Asian Restaurant-Japanese Restaurant-Sushi Restaurant*) from the root (i.e., *Food*) to the leaf node (i.e., *Sushi Restaurant*) have similar semantic properties, and such information could help us capture the semantic relationships between categories. How to explore such structure to guide the venue category representation learning is largely untapped in the literature. (2) **Utilizing multiple types of context**: Existing venue embedding methods mainly consider simple linear contexts and model the co-occurrence patterns of venues based on their contexts. Many different types of context (e.g., dependency-based context and linear context) have been explored in traditional sparse vector-space models [7, 16, 18]; however, it is unclear what contexts are useful in our task and how they can be systematically incorporated into an embedding model.

To address the aforementioned challenges, we consider a holistic approach to venue category representation learning that takes the best of both data modalities (category sequences and category hierarchy) simultaneously. A first attempt is to build a model that integrates one model learned from each of the aforementioned data modalities using linear combination. However, the two individual models cannot capture the sequential and hierarchical information from two modalities collaboratively, defeating the purpose of this combined model. Therefore, we propose a venue **C**ategory **E**mbedding **M**odel named Hier-CEM by adding a newly designed collaborative component for encoding the **Hier**archical structure of categories with these category sequences. Specifically, Hier-CEM consists of two components: category sequence embedding and category hierarchy embedding. In the first component, we capture the sequential relatedness between a category and its neighbors using venue category sequences. The second component aims to improve the representations of categories by integrating hierarchical category information. To be specific, we include more semantically relevant categories (parent categories of each category) using the category hierarchy and establish connections between categories in the sequence and categories in the hierarchy and regulate distributional venue category semantics accordingly. Finally, we perform parameter inference through the stochastic gradient descent method.

To evaluate how well the proposed Hier-CEM could capture the sequential and semantic properties in learning the representations of venue categories, we perform three tasks, including venue semantics study, next category prediction, and venue recommendation, on two real check-in datasets collected from Foursquare. We show that by modeling the category hierarchy and the category sequences collaboratively, Hier-CEM outperforms the state-of-the-art embedding methods on these evaluation tasks.We also provide qualitative analysis on the category embeddings and conclude that Hier-CEM learns category representations that separate different categories better than the baselines.

The main contributions of this article are as follows:

- The proposed Hier-CEM is a holistic approach to venue category representation learning that is able to simultaneously leverage the behavioral data (check-in sequences) and prior knowledge (category hierarchy), whereas existing methods only model the check-in sequences without taking advantage of the wealth of information encoded by the category hierarchy.
- The proposed Hier-CEM establishes connections between category sequences and category hierarchy by adding a newly designed collaborative category hierarchy embedding component. In this way, Hier-CEM benefits from both the linear contextual information as well as the hierarchical relations to learn the category embeddings.
- We experiment Hier-CEM on two public check-in datasets and evaluate its performance using three tasks. Experimental results show that Hier-CEM demonstrates significant performance gains over the baseline methods according to the *superiority paired t-test* and the category hierarchy embedding component could help learn better category embeddings.

The remainder of this article is organized as follows: Section 2 reviews related work. Section 3 gives the preliminaries of our work. Section 4 describes our approach of representation learning for categories. Section 5 introduces three tasks using the venue category embeddings. The experimental results and performance analysis are presented in Section 6. Section 7 concludes this article.

2 RELATED WORK

Our work is related to a broad spectrum of check-in data mining including POI embedding learning, venue category embedding learning, and hierarchical text embedding learning.

2.1 Point-of-Interest Embedding Learning

Most research on representation learning with check-in data originates from word embedding techniques (e.g., Word2vec) [15, 17, 20, 26, 27]. There are two types of Word2vec models: **Continuous-Bag-of-Words (CBOW)** and SkipGram. CBOW models word embeddings by predicting a target word given contextual words, whereas SkipGram does it the other way around. Many studies adopt the framework of Word2vec to construct check-in embeddings [8, 23, 46, 48] by modeling user preference, venue sequences, and time. For example, Zhao et al. [46] assume that the contextual check-in information implies complementary knowledge of POIs and propose a Geo-Temporal sequential embedding rank (Geo-Teaser) model for POI recommendation. Specifically, they capture the check-ins' sequential contexts, the various temporal characteristics, and the geographical influence to learn venue embeddings. Liu et al. [23] model the check-in sequences based on SkipGram. They consider the confidences of observed user preferences for venues with a pairwise ranking loss and leverage the representations for personalized venue recommendations. Feng et al. [8] present a representation model named POI2Vec that captures user preference, location sequential transition influence, and geographical influence for predicting potential visitors for a given location. Zhao et al. [48] propose a Time-Aware Trajectory Embedding Model that considers surrounding locations, dynamic user preference, and time. Specially, they jointly model multiple kinds of temporal factors in a unified manner.

Pushing further from the check-in data, there are some studies that improve the quality of check-in embeddings by taking external information (e.g., social relations, taxi trip data, and tag words of locations) into account. For example, Zhang and Chow [45] exploit the geographical, social, and categorical correlations learned from historical check-ins to predict the relevance score of a user to an unvisited POI to make POI recommendation for users. Yang et al. [37] propose a hypergraph embedding approach (named LBSN2Vec) by revisiting user mobility and social relationships in LBSNs. LBSN2Vec performs random-walk-with-stay to jointly sample user mobility patterns and social relationships from the LBSN hypergraph, and then learns node embeddings from the sampled hyperedges. Zhou and Huang [51] propose an embedding model (DeepMove) to learn the latent representations of locations. They take New York taxi trip data as input and learn location representations, which can incorporate the spatial and temporal semantics of locations and be used for analyzing location similarity and relationships. Xie et al. [35] develop a graph-based embedding model to capture the sequential effect, geographical influence, temporal cyclic effect, and semantic effect in a unified way by embedding the four corresponding relational graphs (POI-POI, POI-Region, POI-Time, POI-Word) into a shared low-dimensional space. Aliannejadi et al. [2] leverage users' ratings on venues to compute a personalized ranking function for each user to rank relevant venues higher than irrelevant ones and learn the latent vectors of users and venues based on matrix factorization accordingly. Further, they [1] propose a personalized keyword boosting method (called PK-Boosting) that models user tags and venue taste keywords to address the data sparsity problem for venue recommendation. However, the above embedding methods can only be applied to some specific check-in data with these external information.

In addition, there also exist some methods using the recurrent neural networks to model the sequential patterns of check-in sequences, in which the venue embeddings can be learned as by-products. For example, Kong and Wu [14] propose a hierarchical spatio-temporal **Long Short-Term Memory (LSTM)** model for venue prediction. They introduce spatial-temporal factors into gate mechanism and leverage historical visit information based on a hierarchical architecture to boost the prediction performance. Manotumruksa et al. [25] leverage RNN-based framework to model both sequence of check-ins and the contextual information (e.g., reviews and time) to capture users' dynamic preference on venues. Yu et al. [43] propose a **category-aware deep model (CatDM)** for next POI prediction using the sparse check-in data. Specifically, they design two

LSTM encoders to model the time series of POI categories and the temporal patterns to capture user preferences in POIs, and then consider multiple correlations to calculate the probability of each POI in the candidate set.

2.2 Venue Category Embedding Learning

Further, some studies consider venue categories in check-in sequences and learn multiple check-in attribute representations including venue category representations, which are more similar to our work. Some methods apply matrix/tensor factorization on user-category matrix to learn the latent vectors of categories. For example, Liu et al. [22] first cluster users based on the similarity of their check-in behavior and then apply matrix factorization to learn the extent of a user's preference on category transitions and finally make category-aware POI recommendation. Further, He et al. [9] decompose a third-rank transition tensor (where each element represents the observed transition frequency of a user from a category to another) by a **Listwise Bayesian Personalized Ranking (LBPR)** approach to learn the latent vectors for users and next categories, and then make next category prediction accordingly. In addition, there exist some methods that leverage word embedding techniques to learn category representations. For instance, Zhou et al. [50] propose a general **Multi-Context Trajectory Embedding Model (MC-TEM)**, which leverages multiple contexts including users, trajectories, surrounding locations, and their categories to predict the target location. MC-TEM projects all the context information into the same embedding space and learns embedding vectors for locations, category labels, user preference, trajectory intent, day and hour, respectively. Yang and Eickhoff [40] present a **Spatio-Temporal Embedding Similarity algorithm (STES)**, which concatenates a venue category and its check-in time slot as a feature word and builds a feature word sequence for each user according to his/her check-ins. Then STES trains a vector representation for each feature word using the CBOW model. Yan et al. [36] propose an approach, namely, augmented **spatial contexts (SCs)**, to capture the semantics of venue categories by modeling category embeddings based on the assumption that venues can be categorized by their neighbors. However, these methods just model the linear context of check-in sequences without taking into account the hierarchical structure of venue categories.

2.3 Hierarchical Text Embedding Learning

We incorporate the hierarchical structure of venue categories into the embedding method to learn venue semantics. This is related to a body of literature on hierarchical text embedding. For example, Hu et al. [11] train a distributed representation for the whole entity hierarchy of Wikipedia based on the assumption that far-away entities in the hierarchy tend to be semantically distant and nearby entities tend to share common semantic features. Liu et al. [21] propose a semantic **structure-based word embedding method (SENSE)** that incorporates the constraints of concept convergence and word divergence into the Word2vec model. Specifically, they assume that a word should be close to the center of words on the lower level and far away from those words in the same level in the structure of WordNet. Alsuhaibani et al. [3] present a **Hierarchical Word Embedding (HWE)** method for identifying the hypernymy relations between words. HWE encodes not only the direct hypernymy relations between the hypernym and hyponym words, but also the indirect and the full hierarchical hypernym path. Zhou et al. [49] propose a **hierarchy-aware global model (HiAGM)** that extracts the label structural information to aggregate the label-wise text features for hierarchical text classification. HiAGM consists of a traditional text encoder for extracting textual information and a hierarchy-aware structure encoder for modeling hierarchical label relations. In contrast to the methods mentioned above that mainly model the relations between nodes in the hierarchical structure, the proposed Hier-CEM models the linear context (from check-in sequences) and the hierarchical context (from category hierarchy) collaboratively.

Table 1. Notations and Descriptions

Notations	Descriptions
u, v, t, c	user, venue, time, and venue category
s, N_s	venue category sequence, length of s
$\mathcal{C}$	set of venue categories
$\mathcal{S}$	set of venue category sequences
$\mathcal{H}$	venue category hierarchy
$\mathcal{A}(c)$	ancestor categories of c
$\mathbf{v}_c$	vector of a context category c
$\mathbf{v}'_c$	vector of a target category c
k	context window size
d	embedding size
N_e	negative sample size

3 PRELIMINARIES

3.1 Formalization

We first define the concepts and the problem, and then list the notations and their descriptions in Table 1.

Venue category hierarchy. Without loss of generality, we adopt the venue categories used by Foursquare, which constitute a five-layer hierarchical structure, as the category hierarchy.

The entire tree can be viewed here:[1] The top layer includes 10 categories, i.e., *Arts & Entertainment*, *College & University*, *Event*, *Food*, *Nightlife Spot*, *Outdoors & Recreation*, *Professional & Other Places*, *Residence*, *Shop & Service*, *Travel & Transport*.

Check-in. A check-in is defined as a tuple $<u, v, c, t>$ that depicts that a user u visits a venue v at time t, where c demonstrates the category of the visited venue, which could be at any layer of the category hierarchy.

Venue category sequences. Given a user, we first sort all his/her check-ins over a (configurable) period of time according to the visited time and obtain a check-in sequence. Then, we use the category c in each check-in tuple to construct a chronologically ordered sequence of venue categories.

Given *venue category sequences* and *venue category hierarchy*, our goal is to learn embedding vectors of venue categories in a latent semantic space.

3.2 Data Preparation

Check-in data. The check-in data are from Foursquare, including long-term (about 18 months from April 2012 to September 2013) check-ins collected from the United States and Japan [38, 39]. Each check-in contains *user ID*, *venue ID*, *venue category,* and *timestamp*. To reduce noises, we remove those users who have less than 50 check-ins and those categories that occur less than 100 times. After this pre-processing, the data collected from Japan contain 2,204,192 check-ins by 10,336 users and those from the United States contain 3,182,412 check-ins by 21,898 users. Finally, we sort all the check-ins of each user and obtain 10,336 venue category sequences containing 330 venue categories as the JP dataset and 21,898 category sequences containing 398 categories as

[1] https://developer.foursquare.com/docs/resources/categories.

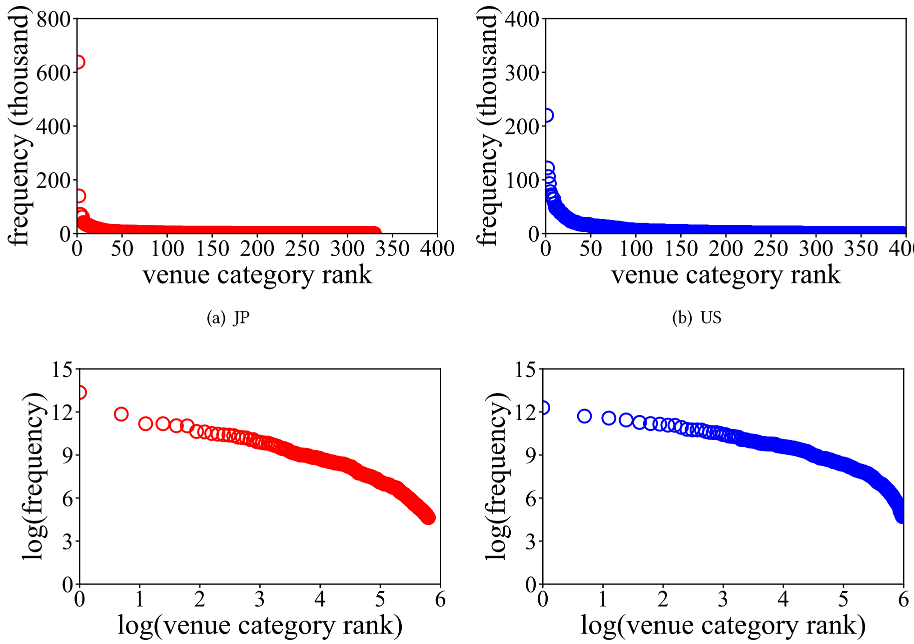


Fig. 1. Venue category rank-frequency and log-log plot.

the US dataset. The average length of venue category sequence is 213 in the JP dataset and 145 in the US dataset, respectively.

After analyzing the venue categories and their frequencies in JP and US datasets, we see a long tail in the rank-frequency distribution, as shown in Figure 1. The log-log plot also shows a linear trend. Fitting log(frequency) and log(rank) using the linear regression, yields values of 0.9208 and 0.8421 for R-squared in JP and US datasets, respectively, which indicates that the model fits strongly to the data. Simply put, these statistics show that the rank-frequency indeed follows a power law distribution by which a few venue categories dominate the data. This is an important motivation for the proposed category embedding method discussed in the following section.

Venue category hierarchy. Foursquare organizes its venue categories into a five-layer hierarchical structure, and we adopt it as the venue category hierarchy. The top-layer of this structure contains 10 non-leaf nodes (coarse venue categories). To visualize the coverage of categories in JP and US datasets, we plot and compare the distribution curves over the number of leaf categories between our datasets and the original structure, as shown in Figure 2. It is worth mentioning that the number of leaf categories distributed in both our datasets and Foursquare is unbalanced. For example, the *Food* category in US dataset contains 86 leaf categories, while the *Event* category does not have any leaf nodes. The top two first-layer categories are *Food* and *Shop & Service* in our datasets.

However, we observe that some leaf categories contain only a small number of check-ins. In our datasets, we have removed those categories with less than 100 check-ins. Tables 2 and 3 list the top five leaf categories with the most and the least check-ins, respectively. To tackle the issue of sparseness, we leverage the category hierarchy to build the augmented hierarchical contexts (detailed in Section 4.3).

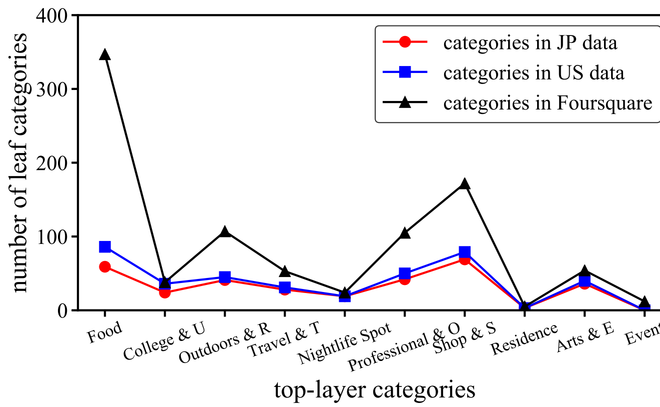


Fig. 2. Top-layer venue category distribution in terms of the number of their leaf categories.

Table 2. Leaf Categories with the Most and the Least Check-ins (JP)

Leaf Category with the Most Check-ins	Num.	Leaf Category with the Least Check-ins	Num.
Train Station	638,726	*Campground*	109
Metro Station	140,984	*College Residence Hall*	108
Noodle House	72,336	*Dance Studio*	106
Convenience Store	71,819	*Eastern European Restaurant*	104
Shopping Mall	62,654	*Planetarium*	103

Table 3. Leaf Categories with the Most and the Least Check-ins (US)

Leaf Category with the Most Check-ins	Num.	Leaf Category with the Least Check-ins	Num.
Home (private)	220,710	*Swiss Restaurant*	127
Office	122,003	*Gluten-free Restaurant*	119
Coffee Shop	106,287	*College Soccer Field*	118
Bar	93,533	*Distillery*	115
Grocery Store	78,809	*College Hockey Rink*	110

4 THE PROPOSED HIER-CEM

In this section, we describe the category sequences embedding method and how to improve category representations by integrating the hierarchical structure of venue categories. The embedding method originates from natural language processing and has been used successfully in many domains. By acknowledging the similarity between venue category sequences and linguistic expressions, we first model the linear context from the category sequences directly to learn category embeddings. We then introduce two approaches to model the category hierarchy in determining latent category representations. Finally, we propose the parameter learning algorithm for the proposed Hier-CEM.

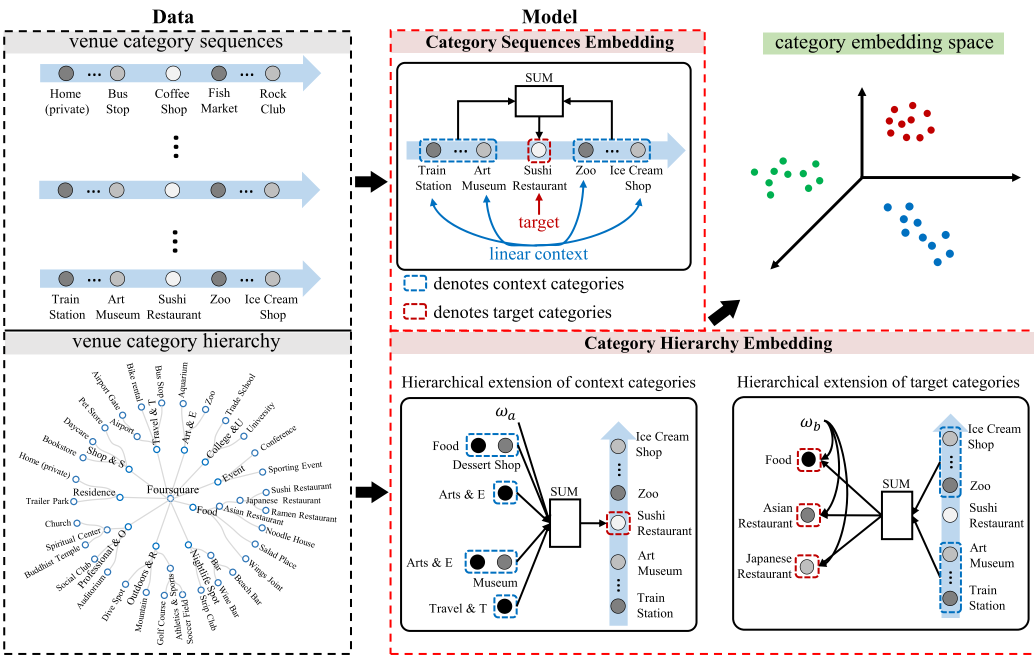


Fig. 3. The framework of Hier-CEM.

4.1 Overview

To learn representations of venue categories, we consider two data modalities: venue category sequences from users' check-ins and the pre-defined venue category hierarchy, as shown in the left part of Figure 3. The category sequences imply users' mobility patterns while the category hierarchy contains the hierarchical semantic relations among categories. Based on the two kinds of data, we build an embedding model, which comprises two components: category sequences embedding and category hierarchy embedding. In the first component, we model the relation between the given category and its linear context. However, some categories are so sparse in the category sequences (as reported in Tables 2 and 3) that it is difficult to yield meaningful latent representations via modeling the linear context. Meanwhile, venue categories are organized with a hierarchical structure, and their semantics are hierarchically correlated. Therefore, we design a new collaborative component that embeds the hierarchical structure of categories. We separately construct two kinds of hierarchical context using the category hierarchy and establish connections between categories in the sequence and categories in the hierarchy. Finally, the proposed Hier-CEM embeds venue categories into the latent space, where semantically related categories ought to be close to each other, as shown in the upper-right part of Figure 3.

4.2 Category Sequences Embedding

Recent work has shown that the word embedding models (e.g., CBOW) can effectively capture the semantic relationships from word sequences based on the distributional semantics assumption [18, 20, 26, 40]. From analyzing the venue category distribution of category sequences, we know that, similarly to the word frequency distribution of texts, it follows a power law distribution (as introduced in Section 3.2). This leads us to taking advantage of the embedding method and the underlying distributional semantics assumption for the study of category representations.

As venue categories are the summarization of the semantic information of venues, we directly learn representations of venue categories by modeling the category sequences. We select the CBOW model, which predicts the target category c_i given the preceding and following categories in a sliding window from c_{i-k_1} to c_{i+k_2}, where $k = k_1 + k_2$ is the context window size. For example, as shown in the left-middle part of Figure 3, given the category sequence, we take *Sushi Restaurant* as the target category, and the preceding and following categories (e.g., *Train Station*, *Art Museum*, *Zoo*, *Ice Cream Shop*) as the context categories.

Our objective is to approximate the true check-in category probability distribution from the data. A typical approach is to use cross entropy to measure the difference between the learned probability and the true probability. Since our data is discrete and we only care about the target category, the cross entropy for one target category c_i can be simplified as

$$D(\hat{y}_{c_i}, y_{c_i}) = -y_{c_i} \log \hat{y}_{c_i}, \tag{1}$$

where $\hat{y}_{c_i}$ is the predicted probability of the target category c_i given the linear context, and y_{c_i} is the true probability of c_i given the context. $\hat{y}_{c_i}$ can be further defined as

$$\hat{y}_{c_i} = p(c_i | c_{i-k_1}, \ldots, c_{i+k_2}). \tag{2}$$

To calculate the probability $\hat{y}_{c_i}$, we leverage the softmax function to turn the scores into probabilities and substitute the categories with vector representations. Note that y_{c_i} will always be 1. For simplicity, we use c_{ik} to represent the context categories $c_{i-k_1}, \ldots, c_{i+k_2}$ in the following description. It is formally defined as,

$$\min \sum_{s \in \mathcal{S}} \sum_{i=1}^{N_s} - \log p(c_i | c_{ik}), \tag{3}$$

where

$$\begin{aligned} p(c_i | c_{ik}) &= \frac{exp\left(\mathbf{v}'_{c_i} \cdot \mathbf{v}_{c_{ik}}\right)}{\sum_{j=1}^{|C|} exp\left(\mathbf{v}'_{c_j} \cdot \mathbf{v}_{c_{ik}}\right)}, \\ \mathbf{v}_{c_{ik}} &= \frac{1}{k}\left(\mathbf{v}_{c_{i-k_1}} + \cdots + \mathbf{v}_{c_{i+k_2}}\right). \end{aligned} \tag{4}$$

In Equations (3) and (4), $\mathbf{v}'_c$ and $\mathbf{v}_c$ are the target category vector and the context category vector, respectively; $|C|$ is the total number of venue categories; $\mathcal{S}$ is the set of venue category sequences and N_s is the length of the category sequence s. In the objective (Equation (3)), the relations between the linear context and the target category are modeled, and categories with similar linear context tend to be close in the embedding space.

4.3 Category Hierarchy Embedding

In addition to the category sequences, venue categories can be naturally grouped in a tree-like hierarchy based on semantics or domain knowledge. The hierarchy contains multiple layers and categories from the root to the leaf node usually have similar semantics. The representations of categories obtained via modeling linear context of venue category sequences do not take into account the semantics of the category hierarchy. Therefore, we propose to improve the category representations by integrating the category hierarchy. In particular, we build two types of hierarchical context to model the co-occurrence patterns between the context and the target to better learn representations of venue categories.

4.3.1 Hierarchial Extension of Context Categories. To model the hierarchy of venue categories, we propose to generalize the concept of "context." In the component of category sequences embedding, we consider the preceding and following categories of the target as the linear context. Here, we propose the hierarchical context to encode the hierarchical structure of categories. In a category hierarchy, the categories at lower layers cover more specific semantics than categories at upper layers. Based on this observation, for each category in the linear context, we take all its ancestors (from the root to itself) as the hierarchical extension of the context category. For instance, as depicted in the bottom part of Figure 3, for each context category (e.g., *Art Museum*), we find all its ancestor categories (i.e., *Museum* and *Art & Entertainment*) via the venue category hierarchy. Evidently, the newly constructed hierarchical extension of context categories shares similar semantics with the linear context and is able to include relevant categories that are far away from the target in the venue category sequences.

We predict the target category (e.g., *Sushi Restaurant*) given its hierarchical context (i.e., hierarchical extension of context categories). Intuitively, categories from different layers are included in the hierarchical context, and an increasing distance of the ancestor category from the context category in the hierarchy would decrease the power of the ancestor category in predicting the target category. For example, *Art Museum* is more relevant to *Museum* than *Art & Entertainment*, and *Museum* ought to play a more important role in predicting the target *Sushi Restaurant*. Hence, we add weights (i.e., ω_a in Figure 3) for the ancestor categories in the hierarchical context and define the objective as

$$\min \sum_{s\in\mathcal{S}} \sum_{i=1}^{N_s} -\log p(c_i|\mathcal{A}(c_{ik})), \tag{5}$$

where $\mathcal{A}(c_{ik})$ represents the hierarchical extension of context categories c_{ik}.

We compute $p(c_i|\mathcal{A}(c_{ik}))$ with the softmax function,

$$\begin{aligned} p(c_i|\mathcal{A}(c_{ik})) &= \frac{exp\left(\mathbf{v}'_{c_i} \cdot \mathbf{v}_{\mathcal{A}(c_{ik})}\right)}{\sum_{j=1}^{|C|} exp\left(\mathbf{v}'_{c_j} \cdot \mathbf{v}_{\mathcal{A}(c_{ik})}\right)}, \\ \mathbf{v}_{\mathcal{A}(c_{ik})} &= \sum_{c_a \in \mathcal{A}(c_{ik})} \omega_a \mathbf{v}_{c_a}, \\ \omega_a &\propto \frac{1}{dis(c_{ik}, c_a)} N(c_a), \quad \sum_a \omega_a = 1, \end{aligned} \tag{6}$$

where ω_a is the weight of each ancestor category in predicting the target category, $dis(c_{ik}, c_a)$ denotes the number of steps going down from the ancestor category c_a to the corresponding context category in the given hierarchy, and $N(c_a)$ is the number of category c_a acting as the ancestor of the context categories. With the weighted function ω_a, we guarantee that a category is more relevant to its closer ancestor.

4.3.2 Hierarchial Extension of Target Categories. Further, we propose to build the hierarchical extension of target categories, i.e., the ancestor categories of the target category, as the other kind of hierarchical context. As categories along the same vein in the category hierarchy have similar semantic meanings, the newly designed hierarchical extension of target categories could include semantically related categories that are far away from the contextual categories in the category sequences. We model the co-occurrence information of the linear context and any category in the hierarchical extension. For example, as shown in the bottom-right part of Figure 3, we first find the ancestors (i.e., *Japanese Restaurant*, *Asian Restaurant*, and *Food*) of the target category *Sushi Restaurant* based on the category hierarchy, and then predict each of them given the linear context (e.g., *Train Station*, *Ice Cream Shop*). Note: A distance decay function is also considered here (i.e., ω_b in Figure 3). We formulate such learning as follows:

$$\min \sum_{s \in \mathcal{S}} \sum_{i=1}^{N_s} \sum_{c_b \in \mathcal{A}(c_i)} -\omega_b \log p(c_b | c_{ik}), \tag{7}$$

where

$$\omega_b \propto \frac{1}{dis(c_i, c_b)}, \quad \sum_b \omega_b = 1. \tag{8}$$

$\mathcal{A}(c_i)$ represents the hierarchical extension of the target category c_i, ω_b is the weight of each target ancestor category, and $dis(c_i, c_b)$ denotes the number of steps going down from the ancestor category c_b to the target category c_i.

Given venue category sequences and the venue category hierarchy, we jointly model different types of context (including linear context and two kinds of hierarchical context) and minimize the overall objective as follows:

$$\begin{aligned}\min_{\mathbf{v}, \mathbf{v}'} \sum_{s \in \mathcal{S}} \sum_{i=1}^{N_s} [&- \log p(c_i | c_{ik}) - \log p(c_i | \mathcal{A}(c_{ik})) \\ &- \sum_{c_b \in \mathcal{A}(c_i)} \omega_b \log p(c_b | c_{ik})] + \frac{\lambda}{2} \left(||\mathbf{v}||_2^2 + ||\mathbf{v}'||_2^2 \right),\end{aligned} \tag{9}$$

where λ is the regularization parameter, and $|| \cdot ||_2$ is the 2-norm.

4.4 Parameter Learning

The parameters of Hier-CEM include the vectors of context categories $\mathbf{v}_c$ and target categories $\mathbf{v}'_c$. For parameter learning, we need to minimize the objective defined in Equation (9). However, it is impractical to directly optimize this objective, because the cost of computing the full softmax to predict the target category is extremely high. Hence, we adopt the efficient and effective negative sampling strategy to approximate the full softmax [18, 28, 33, 34]. When training the vector of c_i, we first obtain a negative sample set $\mathcal{N}(c_i)$, in which c_x is not the same as c_i if $c_x \in \mathcal{N}(c_i)$. The negatively sampled category c_x is randomly selected on the basis of its unigram distribution $(\frac{\sharp(c)}{\sum_c \sharp(c)})^{ds}$, where $\sharp(c)$ is the number of times that category c appears in the training set, and ds is the distribution smoothing hyper-parameter, which is usually defined as 0.75. Then, we define $F^{c_i}(c) = 1$ if $c = c_i$, and $F^{c_i}(c) = 0$ otherwise, where $F^{c_i}(c)$ is the label of the category c.

Given the context of a target category c_i, we maximize the occurrence probability of c_i and meanwhile minimize the occurrence probability of negative samples $c_x \in \mathcal{N}(c_i)$. Therefore, we can replace $\log p(c_i | c_{ik})$ with

$$\log \prod_{c \in c_i \bigcup \mathcal{N}(c_i)} p(c | c_{ik}), \tag{10}$$

where

$$p(c | c_{ik}) = \begin{cases} \sigma(\mathbf{v}'_c \cdot \mathbf{v}_{c_{ik}}), & F^{c_i}(c) = 1; \\ 1 - \sigma(\mathbf{v}'_c \cdot \mathbf{v}_{c_{ik}}), & F^{c_i}(c) = 0, \end{cases} \tag{11}$$

in which $\sigma(z) = (1 + \exp(-z))^{-1}$ is the sigmoid function.

ALGORITHM 1: Hier-CEM ($\mathcal{S}$, $\mathcal{H}$, k, d, N_e)

```
Require: venue category sequence set S
     venue category hierarchy H
     context window size k
     embedding size d
     negative sample size N_e
Ensure: context category representations v_c ∈ R^d
     target category representations v'_c ∈ R^d
1:  initialize v_c and v'_c using a Gaussian distribution
2:  for each category sequence s ∈ S do
3:     for each target category c_i in s do
4:        get the linear context c_ik
5:        get the hierarchical extension A(c_ik) of c_ik using H
6:        get the hierarchical extension A(c_i) of c_i using H
7:        sample N_e negative target categories c_x
8:        update v_c and v'_c, where v_c = v_c − η ∗ ∂L/∂v_c and v'_c = v'_c − η ∗ ∂L/∂v'_c
9:     end for
10: end for
```

Further, with a series of simplifications, we reach the final objective function L as follows:

$$
\begin{aligned}
\min_{\mathbf{v},\mathbf{v}'} \sum_{s\in\mathcal{S}} \sum_{i=1}^{N_s} & -[\log\sigma\left(\mathbf{v}'_{c_i}\cdot\mathbf{v}_{c_{ik}}\right) + \sum_{c\in\mathcal{N}(c_i)} \log\sigma\left(-\mathbf{v}'_c\cdot\mathbf{v}_{c_{ik}}\right)] \\
& - [\log\sigma\left(\mathbf{v}'_{c_i}\cdot\mathbf{v}_{\mathcal{A}(c_{ik})}\right) + \sum_{c\in\mathcal{N}(c_i)} \log\sigma\left(-\mathbf{v}'_c\cdot\mathbf{v}_{\mathcal{A}(c_{ik})}\right)] \\
& - \sum_{c_b\in\mathcal{A}(c_i)} \omega_b[\log\sigma\left(\mathbf{v}'_{c_b}\cdot\mathbf{v}_{c_{ik}}\right) + \sum_{c\in\mathcal{N}(c_b)} \log\sigma(-\mathbf{v}'_c\cdot\mathbf{v}_{c_{ik}})].
\end{aligned}
\tag{12}
$$

We train the proposed Hier-CEM using the stochastic gradient descent method and show the pseudo code in Algorithm 1. Lines 2–10 in Algorithm 1 show the core of our Hier-CEM. In the two loops, we traverse each target category of sequences from the category sequence dataset. For each target category c_i, we first get the linear context c_{ik}, the hierarchical extension $\mathcal{A}(c_{ik})$ of each context category c_{ik}, and the hierarchical extension $\mathcal{A}(c_i)$ of each target category c_i, based on the venue category hierarchy, and compute the weight of each category in $\mathcal{A}(c_{ik})$ and $\mathcal{A}(c_i)$, respectively. Then, we sample N_e negative categories for each positive target category. Finally, we use them to update these representations in accordance with our objective function L in Equation (12), where η is the learning rate.

4.5 Complexity Analysis

We provide the complexity analysis of Hier-CEM. When updating $\mathbf{v}_c$ and $\mathbf{v}'_c$, we need to get the linear context and the hierarchical extension for each target category and build the hierarchical extension for each category in the linear context. As the number of venue categories is limited and the hierarchical structure could be stored with a map function, the computational complexity of finding the ancestors of a category could be $O(1)$. Further, the computational complexity of constructing the hierarchical context is $O(k)$, where k is the size of the linear context. In the loops, we traverse each target category of sequences from the category sequence dataset. Hence, the computational complexity of Hier-CEM is $O(k\cdot N_s\cdot|\mathcal{S}|)$, where $|\mathcal{S}|$ is the number of venue category sequences in the training dataset, and N_s is the length of category sequence s. As k is usually a small number, the complexity of Hier-CEM is linear to the size of the training dataset, which is efficient in computation complexity.

5 PROBING VENUE CATEGORY REPRESENTATIONS

We propose surface level tasks to probe the properties encoded in venue category representations, which are important to gain a deeper understanding of the type of information they capture. We split the surface level tasks into two types: the venue semantics study task that focuses on individual venues and the next check-in category prediction task that focuses on check-in sequences. In addition to the two surface level tasks, we propose a downstream prediction task (i.e., venue recommendation) to validate the utility of these category representations.

5.1 Venue Semantics Study

As we claim exploiting the category hierarchy could help to capture the semantic relationships between categories, we first examine whether semantic properties could be encoded in the learned venue category representations. To evaluate how well these category representations encode the semantic properties, we use a metric named **match rate** to measure the semantic overlap of categories [51]. Specifically, for each venue category c_i, we find its nearest neighbor c_j in the embedding space based on the cosine similarity of category representations. If c_i and c_j share the top-layer category in the hierarchy, then there is a match between c_j and c_i, and c_j is considered as a *matched category* of c_i. The match rate of a test set is the ratio of matched categories in this set over its size, i.e.,

$$match\ rate = \frac{number\ of\ matched\ categories}{number\ of\ categories\ in\ the\ test\ set}. \tag{13}$$

Naturally, a higher value indicates that the category representations more distinctly capture semantics.

5.2 Next Category Prediction

Inspired by the methods proposed for making trajectory prediction [5, 43], we propose a next check-in category prediction task, where the goal is to predict the possible next category that a user may visit given his/her current check-in category. This task is to validate whether the category representations could retain the sequential patterns inherent in the check-in sequences.

Given the current category c_i and a candidate category c, we calculate $p(c|c_i) \propto \mathbf{v}'_c \cdot \mathbf{v}_{c_i}$, where $\mathbf{v}'_c$ is the embedding vector of the target category c and $\mathbf{v}_{c_i}$ is the vector of the context category c_i. Based on the values of $p(c|c_i)$, we generate a ranking of categories from the most possible to the least possible. Then, we generate the rankings for all the test instances. We use *accuracy* and *MRR* **(Mean Reciprocal Rank)** to evaluate the performance. *accuracy* is defined as the frequency of the true next category appearing in the top-5 predicted list; *MRR* is defined as

$$MRR = \frac{1}{|C_{test}|} \sum_{i=1}^{|C_{test}|} \frac{1}{rank_i}, \tag{14}$$

where C_{test} is the set of test categories, and $rank_i$ is the rank of the true next category of category c_i in our rankings.

Note that the evaluation of next category prediction is not the same as that of venue semantics study. We utilize the embedding vectors of context categories and compute the cosine similarity in the task of venue semantics study; while we utilize the embedding vectors of both context categories and target categories and compute the dot product in the task of next category prediction.

5.3 Venue Recommendation

In addition to the surface level tasks, we want to know whether the learned category representations could be used in downstream applications. As a case study, we evaluate the learned representations on the task of venue recommendation. The goal of venue recommendation is to predict a list of top-r venues that a specific user may visit in the future. Our recommendation algorithm follows [40], which is based on the user-venue similarity in the newly established embedding space. The recommendation algorithm works as follows:

- **Representing a user.** We represent a user u as $\mathbf{v}_u$ by averaging the embedding vectors $\mathbf{v}_{c_i}$ of all his/her check-in categories and calculate a user coordinate centroid ($coordinate_u$) using those check-in venues ($coordinate_{v_i}$). That is,

$$\begin{aligned} \mathbf{v}_u &= \frac{1}{N_u} \sum_{i=1}^{N_u} \mathbf{v}_{c_i}, \\ coordinate_u &= \frac{1}{N_u} \sum_{i=1}^{N_u} coordinate_{v_i}, \end{aligned} \tag{15}$$

  where N_u is the total number of check-ins of user u, and $coordinate_{v_i}$ is the latitude and longitude of venue v_i.
- **Determining the order of a user's favored categories.** Given a user, we find his/her W favored categories based on the cosine similarities between the user vector $\mathbf{v}_u$ and these category embedding vectors $\mathbf{v}_c$. Here, we list these W favored categories in descending order $f_1, \ldots, f_W$.
- **Recommending venues.** During the recommendation stage, we first select the W most favored categories of the user and focus on unique venues within these selected categories. We then consider both the category preference order f_w and the distance $dist_{uv}$ between a venue v and the user u's coordinate centroid to make recommendation following Reference [40]. Specifically, we use two exponential decay factors, **category decay** CD and **spatial decay** SD, to calculate the final user-venue similarity score (denoted as $Score_{uv}$):

$$\begin{aligned} CD_{uv} &= exp\left(-\omega_1 \times f_w^v\right), \\ SD_{uv} &= exp(-\omega_2 \times dist_{uv}), \\ dist_{uv} &= Distance(coordinate_v, coordinate_u), \\ Score_{uv} &= CD_{uv} \times SD_{uv}, \end{aligned} \tag{16}$$

  where $f_w^v \in \{0, 1, \ldots, W-1\}$ is the user u's preference order corresponding to v's category, $dist_{uv}$ is the Euclidean distance between a venue v and the user coordinate centroid ($coordinate_u$), ω_1 and ω_2 are the weights. Afterwards, given a user, we sort all venues belonging to these W categories in descending order of $Score_{uv}$ and make recommendations from the top.

We evaluate the performance in terms of *precision* and *recall*, which are commonly used in the task of venue recommendation [43, 45]. We denote these metrics at top-r recommendation as *precision@r* and *recall@r*, respectively. The definitions are formulated as follows:

$$\begin{aligned} precision@r &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{V}_g \cap \mathcal{V}_p|}{r}, \\ recall@r &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{V}_g \cap \mathcal{V}_p|}{|\mathcal{V}_g|}, \end{aligned} \tag{17}$$

Table 4. Parameters of Hier-CEM

Parameters	Tested settings
Context window size (k)	2, 4, 6, 8, 10
Embedding size (d)	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Negative sample size (N_e)	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Percentage of training data	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1

where $\mathcal{U}$ is the set of users, $\mathcal{V}_g$ and $\mathcal{V}_p$ denote the set of ground-truth venues and the set of corresponding predicted venues for each user in the test data, and $|\mathcal{U}|$ is the size of set $\mathcal{U}$.

6 EXPERIMENTS

In this section, we begin by introducing the experimental settings; then, we elaborate on the various experiments and present the results of both qualitative and quantitative evaluations.

6.1 Experimental Settings

The parameters we use for the experiments are shown in Table 4. We adopt the adaptive learning rate following the work by Mikolov et al. [26] with the initial learning rate set at $\eta = 0.025$ and the weight of the regularization term at $\lambda = 0.001$. Grid search is employed to select the optimal parameters with a small but adaptive step size.

Baselines. We carry out experiments to compare the overall effectiveness of Hier-CEM with several state-of-the-art category embedding methods:

- **MC-TEM** [50]**:** This is a multi-context trajectory embedding model, in which a venue and its corresponding category are considered as the context and the CBOW model is used to predict the target venue.
- **STES** [40]**:** This is a spatio-temporal embedding similarity algorithm. As we do not include check-in timestamps in this work, we adapt STES by considering the venue category as the feature word and train a vector representation for each feature word with the feature word sequences.
- **SC** [36]**:** The embedding method models the spatial context and the sequential context. Since we do not include the geographic information of venues in this work, we omit the spatial context used in SC for fairness and train the category vectors.
- **LBPR** [9]**:** This is a tensor factorization model, which decomposes a third-rank transition tensor (where each element represents the observed transition frequency of a user from a category to another) by a **Listwise Bayesian Personalized Ranking (LBPR)** approach to learn the latent vectors of categories.
- **CatDM** [43]**:** This is a category-aware deep method that leverages RNNs to model check-in sequences. As we do not include check-in timestamps in this work, we adapt CatDM and use an LSTM encoder to model the sequences of venue categories.
- **SENSE** [21]**:** This is a semantic structure-based word embedding method that combines the semantic constraint (i.e., a category is close to the center of its children and far away from categories on the same level) and the SkipGram using a linear function.
- **Hier-CEM-*context*:** This baseline is a variant of Hier-CEM in that it models the linear context and the hierarchical extension of context categories only. In other words, Hier-CEM-*context* does not incorporate the knowledge of the hierarchical extension of target categories.
- **Hier-CEM-*target*:** This baseline is a variant of Hier-CEM in that it models the linear context and the hierarchical extension of target categories simultaneously.

Table 5. Performance Comparison in Terms of Match Rate on the JP Dataset

Methods	Embedding size (*d*)									
	10	20	30	40	50	60	70	80	90	100
MC-TEM [50]	0.314	0.298	0.345	0.339	0.329	0.342	0.373	0.379	0.382	0.379
STES [40]	0.542	0.549	0.564	0.552	0.552	0.552	0.542	0.536	0.539	0.533
SC [36]	0.509	0.6	0.624	0.621	0.649	0.633	0.624	0.639	0.63	0.639
LBPR [9]	0.215	0.294	0.33	0.397	0.324	0.358	0.427	0.421	0.4	0.409
CatDM [43]	0.594	0.512	0.548	0.545	0.57	0.545	0.603	0.591	0.561	0.585
SENSE [21]	0.658	0.677	0.666	0.671	0.676	0.684	0.672	0.681	0.67	0.69
Hier-CEM-*context*	**0.724**	0.7	0.721	0.73	0.724	0.697	0.706	0.703	0.709	0.706
Hier-CEM-*target*	0.721	0.709	0.742	0.73	**0.755**	0.742	0.724	0.718	0.712	0.712
Hier-CEM	0.703	**0.718**	**0.773**	**0.749**	0.727	**0.749**	**0.755**	**0.749**	**0.727**	**0.73**

Table 6. Performance Comparison in Terms of Match Rate on the US Dataset

Methods	Embedding size (*d*)									
	10	20	30	40	50	60	70	80	90	100
MC-TEM [50]	0.317	0.371	0.356	0.351	0.361	0.4	0.387	0.376	0.4	0.423
STES [40]	0.558	0.634	0.609	0.578	0.558	0.553	0.525	0.508	0.51	0.5
SC [36]	0.548	0.626	0.588	0.599	0.634	0.611	0.634	0.611	0.639	0.631
LBPR [9]	0.225	0.308	0.338	0.333	0.419	0.429	0.379	0.386	0.396	0.364
CatDM [43]	0.679	0.669	0.611	0.641	0.634	0.604	0.646	0.644	0.667	0.659
SENSE [21]	0.722	0.764	0.769	0.774	0.77	0.776	0.771	0.772	0.773	0.779
Hier-CEM-*context*	0.814	0.789	0.761	0.719	0.709	0.694	0.701	0.699	0.691	0.683
Hier-CEM-*target*	0.764	0.781	0.792	0.784	0.764	0.771	0.771	0.769	0.766	0.771
Hier-CEM	**0.816**	**0.854**	**0.851**	**0.849**	**0.846**	**0.851**	**0.816**	**0.843**	**0.838**	**0.851**

6.2 Evaluation on Venue Semantics Study

6.2.1 Comparison with Baselines. The dimensionality, which is the size of an embedding vector, is an important factor that influences the performance of embedding models. As the total number of categories is relatively small compared with the vocabulary size of natural languages, we compare the performance on dimensionalities ranging from 10 to 100 with a step interval of 10. In our experiments, we evaluate all the methods on the same dataset and perform 10 runs to obtain 10 values for the match rate and report the mean of 10 values. Tables 5 and 6 show the match rates with different dimensionalities of all the methods on the JP and US datasets (where the best scores are highlighted in boldface), from which we can make the following observations:

- Compared with MC-TEM, STES and SC yield better accuracies, indicating that directly modeling the category sequences could better capture venue semantics. Recall that MC-TEM, STES, and SC model the linear context to learn category representations. MC-TEM predicts the target venue given multiple contexts (e.g., preceding and following venues, venue categories); it learns the category representation as a byproduct and performs relatively poorly. STES and SC directly model check-in category sequences to generate category representations, with the difference being that STES adopts the CBOW model and SC uses the Skip-Gram model.

- LBPR performs the worst, as it only models users' preference on the category transitions without considering the high-order sequential patterns. CatDM obtains better performances than LBPR. The reason lies in that it applies LSTM-based methods to model long-term transitions in the check-in sequences. Compared with embedding-based methods (e.g., SC), CatDM does not show obvious advantages. For example, it yields worse results than SC on the JP dataset while it outperforms SC on the US dataset in terms of match rate.
- SENSE performs better than those baselines that merely model check-in sequences, because it combines the semantic constraint learned from the hierarchical category structure and the SkipGram model using a linear function, validating that the category hierarchy is vital in capturing venue semantics.
- Hier-CEM performs the best, as it models both the linear contextual information as well as the hierarchical relations to learn the category representations. For example, compared with SC, our Hier-CEM achieves an average improvement of 20% on the JP data and 37.7% on the US data, for various embedding sizes.

Further, we validate whether the performance difference between the proposed Hier-CEM and the baselines is *practically* significant. To show this, we use *superiority paired t-test* to validate whether the difference is indeed significantly greater than a given value M [12, 32]. Suppose we want to evaluate the superiority of a continuous random variable X_T as compared to a second paired random variable X_C. Assume that n paired observations $(X_{Ta}, X_{Ca}), a = 1, 2, \ldots, n$ are available. The D's are the differences formed as $D = X_T - X_C$. Assume that higher values are better, that $\mu_D = \mu_{T-C}$ represents the mean of the differences between the two variables, and that M is the positive superiority margin. The null and alternative hypotheses are

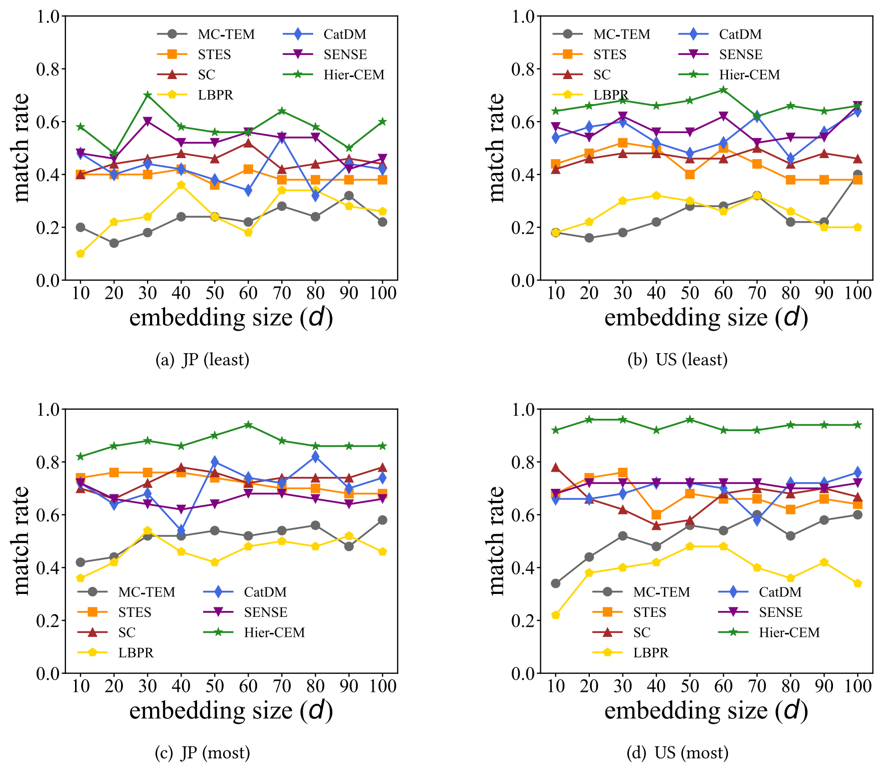
$$\begin{aligned} H0 &: \mu_D \leq M, \\ H1 &: \mu_D > M. \end{aligned} \tag{18}$$

Note that the paired t-test usually tests that the mean difference is zero. The superiority test compares the difference to a non-zero quantity M.

Taking the results of Hier-CEM and MC-TEM given $d = 100$ on the JP dataset as an example, we set $M = 0.32$ and make the null hypothesis that the mean of the differences between the 10 paired observations of Hier-CEM and MC-TEM is less than 0.32. We perform a paired samples t-test on the match rates of Hier-CEM and MC-TEM and observe that the calculated p-value is 0.0038. Therefore, we reject the null hypothesis with this data and conclude that the improvement in match rate by Hier-CEM over MC-TEM, 84.5% (0.32/0.3789), is statistically significant as per the superiority paired t-test with p-value <0.05. Similarly, we perform superiority paired t-tests for Hier-CEM and other baseline methods and conclude that the improvement of Hier-CEM over these baselines in match rate is of practical significance.

6.2.2 Model Analysis. To verify the effectiveness of Hier-CEM, we design two variants: Hier-CEM-*context* and Hier-CEM-*target*. Hier-CEM-*context* incorporates the hierarchical extension of context categories, while Hier-CEM-*target* incorporates the hierarchical extension of target categories. The two variants are to validate the effects of modeling two kinds of hierarchical context simultaneously. We record the comparison results in Tables 5 and 6. From the results, we can find that the performance of the proposed Hier-CEM is better than that of the two variants in most cases.

Further, we validate whether Hier-CEM could learn meaningful representations for those categories with sparse check-ins. Specifically, we perform venue semantics study for the 50 categories with the least check-ins and report their match rates with different embedding sizes for all the methods in Figure 4. Meanwhile, we also report the results on the 50 categories with the most



(a) JP (least)

(b) US (least)

(c) JP (most)

(d) US (most)

Fig. 4. Performance comparison for top-50 categories with the most and the least check-ins.

check-ins. From Figures 4 (a) and (b), we know that Hier-CEM and SENSE perform better than other methods for those 50 categories with the least check-ins, as they involve the hierarchical category structure to solve the problem of category sparseness. Moreover, as can be observed from Figures 4(c) and (d), Hier-CEM outperforms these baselines in the task of venue semantics study for the 50 categories with the most check-ins, indicating that it is effective to model the hierarchical category structure with the category sequences collaboratively to learn category representations. Note that the match rates of Hier-CEM could achieve 0.94 and 0.96 for the 50 categories with the most check-ins on the JP and US datasets, respectively.

6.2.3 Parameter Sensitivity. We analyze the effect of parameters of Hier-CEM on its performance. Since context window size determines the number of contextual venue categories that affect the target category, it will influence the quality of category representations. Figures 5(a) and (b) show the match rates with different context window sizes on the JP and US datasets. With fixed $d = 20, 50$ and $N_e = 1$, the match rates have an obvious improvement when k increases from 2 to 6 for both datasets. When we increase k further, the performance starts to decline, because it takes some unrelated contextual venue categories into consideration. Hence, we set $k = 6$ in the following experiments. We then vary the negative sample size (N_e) from 1 to 10 with $d = 20, 50$, and report the results in Figures 5(c) and (d). We find that the match rates increase gradually as N_e increases from 1 to 5. When N_e is greater than 6, the curve starts to fall. As such, we set $N_e = 5$.

The size of the dataset is also an important factor. To get a good model, we need a certain number of training samples to make sure the model can learn enough information from the data. We select the dataset ranging from 10% to 100% with an increasing step-size of 10% to evaluate the

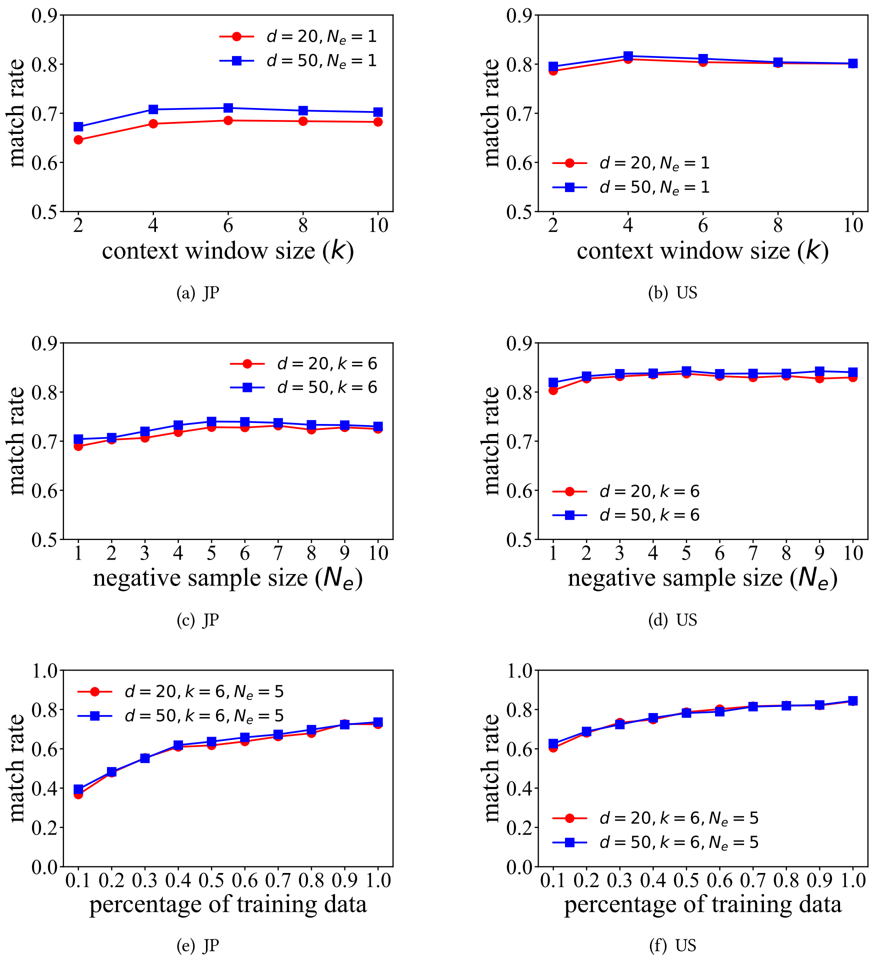
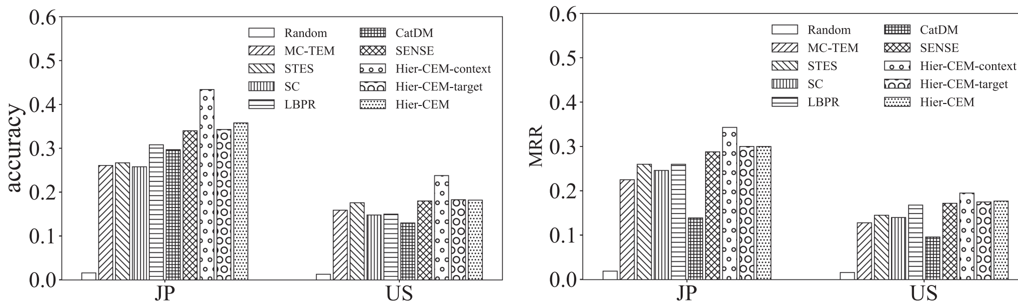


Fig. 5. Effect of varying parameters of Hier-CEM for venue semantics study.

performance. As shown in Figures 5(e) and (f), the match rates on the 10% dataset are only about 0.4 and 0.6 on the JP and US datasets, respectively. As the size of dataset increases, the match rates also increase until the percentage of the dataset reaches 90%, after which the match rates do not change much regardless of data size.

6.3 Evaluation on Next Category Prediction

To evaluate whether the category representations could retain sequential patterns, we design the task of next check-in category prediction. In this experiment, we use 5-fold cross-validation and report the average results. We compare Hier-CEM with the baselines on both JP and US datasets and measure the performance in terms of accuracy and MRR in Figure 6. Here *Random* refers to the method that chooses a category from the category set as the predicted next category, which is regarded as the benchmark. MC-TEM considers both the check-in venues and the corresponding categories as the context and predicts the target venue. As it does not consider the venue category as the target, it performs relatively poorly in this task. SC predicts the context categories given the target category, while STES predicts the target category given the context categories



(a) accuracy (b) MRR

Fig. 6. Performance comparison in the task of next category prediction.

when learning category representations based on the venue category sequences; thus, STES performs better than SC on this task. CatDM obtains poor performances, as it focuses on storing statistical weights for long-term transitions in a check-in sequence, while LBPR generates relatively decent results because it models the category transitions of a user. SENSE models the linear context and the category hierarchy and outperforms the other baselines. It is interesting to observe that Hier-CEM-*context* achieves the best performance. The reason is that incorporating the hierarchical extension of target categories hinders the prediction performance. When incorporating the hierarchical target, we maximize the probability of the ancestor categories of the target category given the context categories. For example, given a category sequence *Train Station* → *Art Museum* → *Sushi Restaurant* → *Zoo* → *Ice Cream Shop*, we find the ancestors (i.e., *Japanese Restaurant*, *Asian Restaurant*, and *Food*) of the target category *Sushi Restaurant* based on the category hierarchy, and maximize the probability of each ancestor (e.g., *Food*) given the linear context (e.g., *Art Museum*, *Zoo*). Therefore, we predict *Food* as the candidate next category of the query sequence *Train Station* → *Art Museum* with a large probability, which results in incorrect sequential prediction, as *Train Station* → *Art Museum* → *Food* does not exist in users' check-in sequences actually. Compared with STES, Hier-CEM-*context* shows an improvement of 31.5% on the JP dataset and 35.2% on the US dataset in terms of accuracy, respectively. Note that the improvements over the baselines are statistically significant using superiority paired t-test with p value <0.01 [12, 32].

6.4 Evaluation on Venue Recommendation

The goal of venue recommendation is to predict a list of top-r venues that a specific user may visit. In addition to the aforementioned baseline methods, we also compare with the following methods for venue recommendation:

- GeoSoCa [45]: This is a venue recommendation method that learns geographical and categorical correlations from the check-in data of users and predicts the relevance score of a user to an unvisited venue.
- PK-Boosting [1]: This is a personalized keyword boosting method that utilizes additional information (e.g., venue keywords, user tags, venue categories) to enhance venue recommendation. As in our evaluation there is no text information on users and venues, we adapt PK-Boosting by considering the venue categories to make venue recommendation.
- CATAPE [31]: This is a category-aware venue embedding method that captures both the sequential influence and categorical information of venues to make venue recommendation.

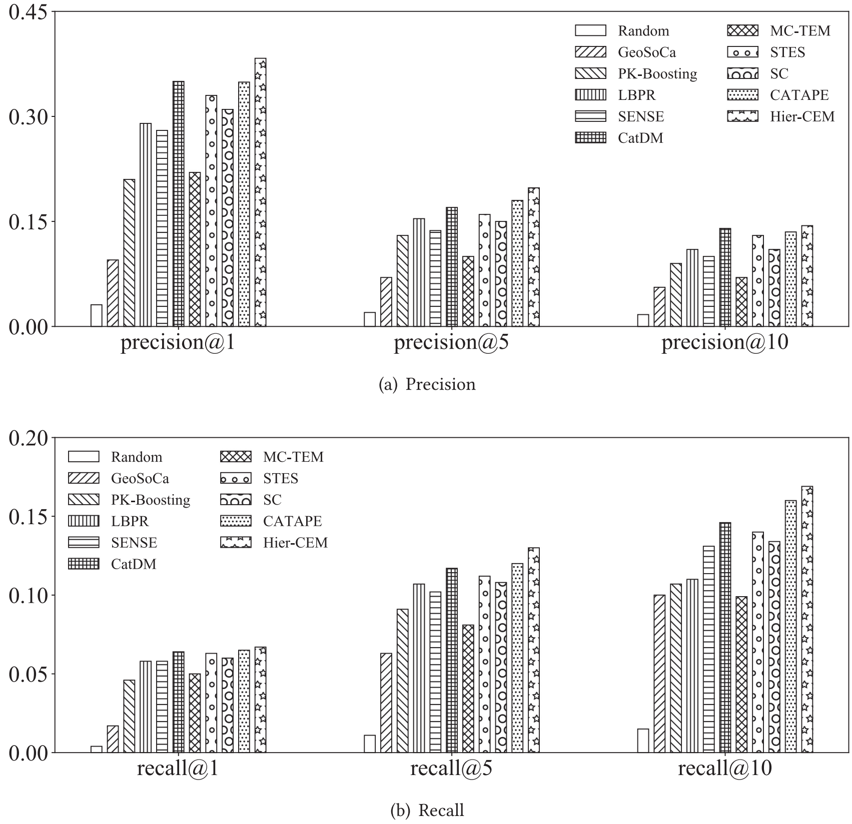
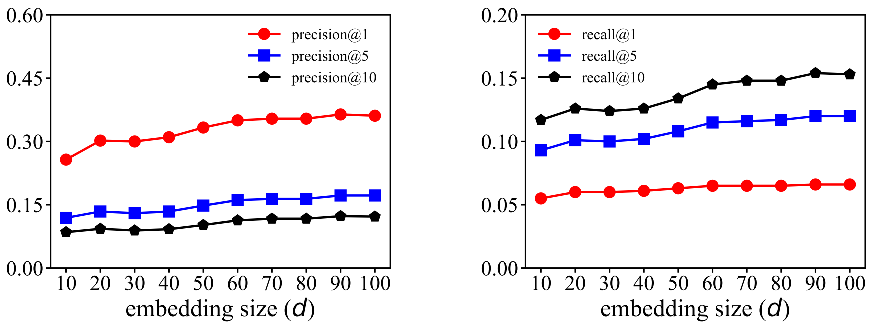


Fig. 7. Performance comparison in the task of venue recommendation.

6.4.1 Comparison with Baselines. We compare the proposed Hier-CEM with these baselines and report the results of all the methods using the US dataset in Figure 7. Similar results are observed on the JP dataset and are omitted here. Note that *Random* means that we randomly select W categories instead of users' preferred ones calculated using category representations as the candidates, which is regarded as the benchmark. We have the following observations based on the results:

- GeoSoCa and PK-Boosting explicitly capture the users' preference over categories using their check-ins without modeling the relations among categories. Therefore, they do not have decent performance compared with those category embedding methods.
- LBPR decomposes a user category transition tensor to learn the representations of categories; SENSE defines two objectives to separately model the check-in sequences and category hierarchy and integrates them simply via a linear function; CatDM models the long-term dependencies in the check-in sequences to learn category representations. They all perform worse than the proposed Hier-CEM.
- MC-TEM performs poorly, as it just considers the category as one of the contexts for predicting the venue. STES and SC directly model the check-in category sequences to learn category representations and perform better than MC-TEM. CATAPE models both the venues and the corresponding categories in the check-in sequences and it outperforms STES and SC.



(a) Precision (b) Recall

Fig. 8. Effect of varying embedding size of Hier-CEM for venue recommendation.

- Hier-CEM performs the best in terms of recall and precision for the task of venue recommendation. The improvements over the baselines are statistically significant using superiority paired t-test with p value <0.01 [12, 32], validating the effectiveness of collaboratively modeling the linear context and the hierarchical context to learn category representations further.

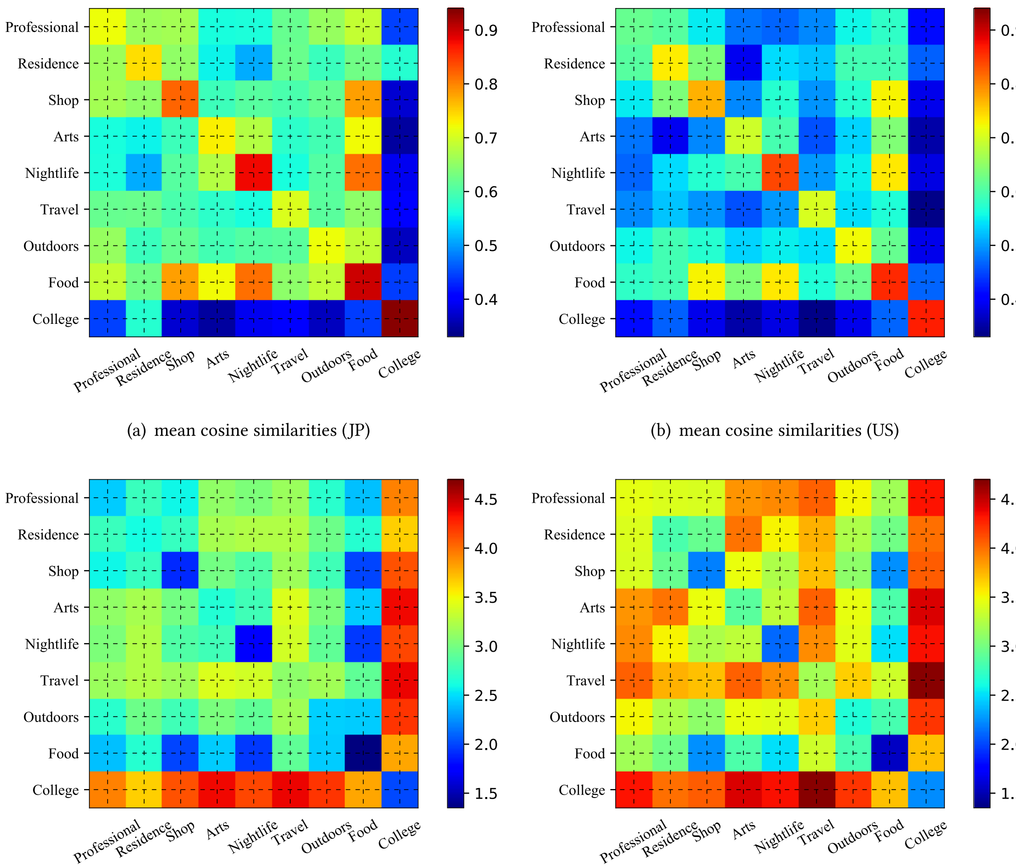
6.4.2 Parameter Sensitivity. We tune the embedding size of Hier-CEM and report the results on venue recommendation using the US dataset in terms of recall and precision in Figure 8. As shown in Figure 8(a), the values of precision improve when we increase the embedding size from 10 to 90, and then remain stable when increasing it further. Similar tendencies in terms of recall are observed from Figure 8(b).

6.5 Qualitative Analysis of Embedding Vectors

Our embedding vectors are designed to retain venue semantics so venue categories are distinguishable. We set the latent embedding dimensionality to 100 and obtain vectors for all the venue categories in the JP and US datasets. 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 measuring the in-between angle and the Euclidean distance demonstrates the magnitude of difference between two embedding vectors. Specifically, we first calculate the cosine similarity and Euclidean distance for each pair of category embedding vectors, and then compute the mean similarity and distance values among the top-layer venue categories. We demonstrate the results in the form of heatmaps in Figure 9, where several significant trends can be observed.

From Figures 9(a) and (b), we can see that the intra-category embedding vectors show the highest mean cosine similarities and that *College* has the largest intra-category cosine similarity and the smallest inter-category similarity. The result appears reasonable, as venues related to *College* are more concentrated in a small district, and their semantics are relatively explicit. We can also observe that categories including similar or overlapping venues have large inter-category similarities, e.g., *Food - Shop & Service* and *Food - Nightlife Spot*, and that category representations learned from the JP dataset show larger intra-category and inter-category similarities than those from the US dataset. Similar tendencies exist with the Euclidean distances between categories in Figures 9(c) and (d).

Further, we select two top-layer categories *Food* and *Outdoors & Recreation* to validate whether their leaf categories are visually distinguishable in a latent space with distributed representations. We take the vectors of these leaf categories from the learned category representations and show



(a) mean cosine similarities (JP)

(b) mean cosine similarities (US)

(c) mean Euclidean distances (JP)

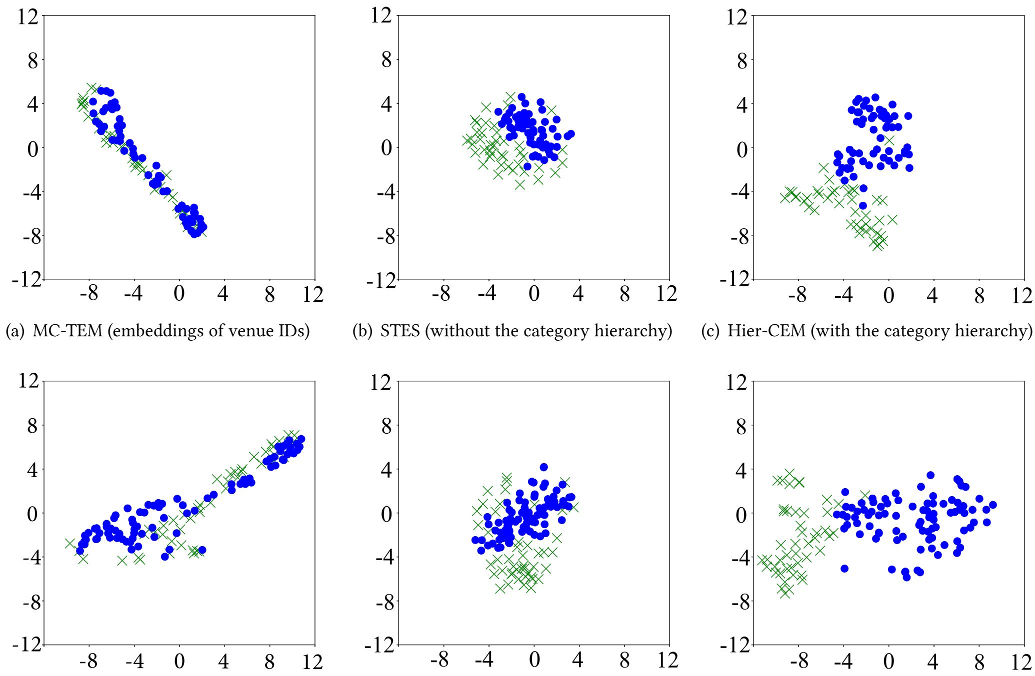
(d) mean Euclidean distances (US)

Fig. 9. Heatmaps of mean cosine similarities and Euclidean distances for venue category embeddings.

them in the 2-dimensional space after dimensionality reduction using **t-Distributed Stochastic Neighbor Embedding (t-SNE)** [24]. The results of applying Hier-CEM and the state-of-the art methods (including MC-TEM [50] and STES [40]) on the JP and US datasets are shown in Figure 10. Evidently, Hier-CEM embeds the leaf categories of *Food* (dots) and *Outdoors & Recreation* (x marks) into two cleanly separated clusters, validating that integrating the hierarchical category information enables learning better semantics than merely modeling the linear context.

Finally, we use several examples to examine whether Hier-CEM can solve the problem of sparse venue categories by involving the hierarchical category structure, as well as how well it captures the semantics of categories with sparse check-ins. Specifically, we perform training of Hier-CEM on the JP and US datasets and launch a bunch of queries using leaf categories with the least check-ins (introduced in Tables 2 and 3). For each query, we retrieve the top-5 most similar categories.

Figure 11 shows the results of MC-TEM, STES and the proposed Hier-CEM when we query with the keywords "*Planetarium*" and "*Campground*" on JP dataset and the keywords "*College Hockey Rink*" and "*Distillery*" on the US dataset. We label the semantically related categories in boldface.



(a) MC-TEM (embeddings of venue IDs) (b) STES (without the category hierarchy) (c) Hier-CEM (with the category hierarchy)

(d) MC-TEM (embeddings of venue IDs) (e) STES (without the category hierarchy) (f) Hier-CEM (with the category hierarchy)

Fig. 10. Hier-CEM learns latent category representations that separate leaf categories of *Food* (denoted by dots) and *Outdoors & Recreation* (denoted by × marks) substantially better than the state-of-the-art MC-TEM [50] and STES [40] models on the JP and US datasets. Here, a standard dimension reduction method t-SNE [24] is used to reduce the 100-dimensional category representations to the 2-dimensional space.

Evidently, MC-TEM performs the worst, as it merely takes the venue category as a kind of context to predict the target venue and is mainly designed to learn representations of venues, difficult to retain the semantics of categories. STES directly models the category sequences and could capture venue semantics decently. But it is not able to learn good representations for those sparse categories. For example, as shown in Figure 11(b), only one of the top-5 results retrieved by STES is related to the query "*Campground*." Hier-CEM embeds the hierarchical structure of categories in the representation learning of categories. Therefore, though some categories are sparse in the linear context, they may be modeled in the hierarchical context and could be trained thoroughly. Observed from Figure 11, we can clearly see that Hier-CEM finds semantically related categories for the given query. For instance, when we query with the keyword "*Distillery*," only Hier-CEM could find the highly relevant category *Brewery*.

7 CONCLUSIONS AND FUTURE WORK

In this article, we present an embedding method (Hier-CEM) to generate a latent representation for each venue category based on two data modalities: venue category sequences and venue category hierarchy. We consider the linear context from venue category sequences and leverage the hierarchical structure of categories to build two kinds of hierarchical context types, and we model the co-occurrence information of categories and their different contexts jointly. These category representations could be used to reason about the similarity among categories. We evaluate the proposed Hier-CEM with two real check-in datasets from Foursquare and compare the embeddings

MC-TEM	STES	**Hier-CEM**
Pizza Place	***Science Museum***	***Aquarium***
Monument / Landmark	*Tapas Restaurant*	***Museum***
Seafood Restaurant	***Art Museum***	***Science Museum***
Garden	*Mediterranean Restaurant*	*Eastern European Restaurant*
Casino	*New American Restaurant*	***Lighthouse***

(a) Query = "*Planetarium*" (JP)

MC-TEM	STES	**Hier-CEM**
Jewelry Store	*Caribbean Restaurant*	***Track***
Deli / Bodega	*Brewery*	***Football Stadium***
Electronics Store	***Athletics & Sports***	***Tennis Court***
Government Building	*New American Restaurant*	***Skating Rink***
Resort	*Nursery School*	***Baseball Field***

(b) Query = "*Campground*" (JP)

MC-TEM	STES	**Hier-CEM**
Market	***College Soccer Field***	***College Soccer Field***
Bagel Shop	***College Basketball Court***	***College Track***
Historic Site	***College Stadium***	***College Stadium***
Monument / Landmark	***Hockey Arena***	***College Baseball Diamond***
Hot Dog Joint	***College Track***	***College Football Field***

(c) Query = "*College Hockey Rink*" (US)

MC-TEM	STES	**Hier-CEM**
Speakeasy	*Roof Deck*	***Pub***
Hot Dog Joint	***Gastropub***	***Brewery***
Clothing Store	***Piano Bar***	***Gastropub***
Pool Hall	*Art Museum*	***Sports Bar***
Nightlife Spot	*Swiss Restaurant*	***Bar***

(d) Query = "*Distillery*" (US)

Fig. 11. Illustrative cases of retrieving top-5 most similar categories for the query categories with sparse check-ins.

generated using Hier-CEM with the state-of-the-art embeddings in the task of venue semantics study, next category prediction, and venue recommendation. The results show that Hier-CEM outperforms the baselines significantly according to the superiority paired t-test and better captures the venue semantics.

Several interesting research problems exist for further exploration. First, we can consider more context types (e.g., spatial context and temporal context) in learning venue category representations. Second, we model category representations with check-ins from Japan and the United States separately, and we could explore the difference between the two groups of representations and how to bridge them in future work.

REFERENCES

[1] Mohammad Aliannejadi and Fabio Crestani. 2018. Personalized context-aware point of interest recommendation. *ACM Trans. Inf. Syst.* 36, 4 (2018), 1–28.

[2] Mohammad Aliannejadi, Dimitrios Rafailidis, and Fabio Crestani. 2018. A collaborative ranking model with multiple location-based similarities for venue suggestion. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*. 19–26.

[3] Mohammed Alsuhaibani, Takanori Maehara, and Danushka Bollegala. 2018. Joint learning of hierarchical word embeddings from a corpus and a taxonomy. In *Automated Knowledge Base Construction*. http://www.akbc.ws/2019/.

[4] Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim, and Jaewoo Kang. 2018. Content-aware hierarchical point-of-interest embedding model for successive POI recommendation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3301–3307.

[5] Meng Chen, Xiaohui Yu, and Yang Liu. 2019. MPE: A mobility pattern embedding model for predicting next locations. *World Wide Web* 22, 6 (2019), 2901–2920.

[6] Meng Chen, Yan Zhao, Yang Liu, Xiaohui Yu, and Kai Zheng. 2020. Modeling spatial trajectories with attribute representation learning. *IEEE Trans. Knowl. Data Eng.* (2020).

[7] Stephen Clark. 2015. Vector space models of lexical meaning. *The Handbook of Contemporary Semantic Theory*. Wiley-Blackwell.

[8] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. POI2Vec: Geographical latent representation for predicting future visitors. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

[9] Jing He, Xin Li, and Lejian Liao. 2017. Category-aware next point-of-interest recommendation via listwise Bayesian personalized ranking. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 1837–1843.

[10] Jiayuan He, Jianzhong Qi, and Kotagiri Ramamohanarao. 2019. A joint context-aware embedding for trip recommendations. In *Proceedings of the 35th International Conference on Data Engineering*. IEEE, 292–303.

[11] Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric Xing. 2015. Entity hierarchy embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. 1292–1300.

[12] David Hull. 1993. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the SIGIR Conference on Research and Development in Information Retrieval*. 329–338.

[13] Krzysztof Janowicz, Martin Raubal, and Werner Kuhn. 2011. The semantics of similarity in geographic information retrieval. *J. Spatial Inf. Sci.* 2011, 2 (2011), 29–57.

[14] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2341–2347.

[15] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*. 1188–1196.

[16] Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. 302–308.

[17] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems*. 2177–2185.

[18] Bofang Li, Tao Liu, Zhe Zhao, Buzhou Tang, Aleksandr Drozd, Anna Rogers, and Xiaoyong Du. 2017. Investigating different syntactic context types and context representations for learning word embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2421–2431.

[19] Yurui Li, Hongmei Chen, Lizhen Wang, and Qing Xiao. 2019. POI representation learning by a hybrid model. In *Proceedings of the 20th IEEE International Conference on Mobile Data Management*. IEEE, 485–490.

[20] Liping Liu, Francisco Ruiz, Susan Athey, and David Blei. 2017. Context selection for embedding models. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*. 4816–4825.

[21] Qian Liu, Heyan Huang, Guangquan Zhang, Yang Gao, Junyu Xuan, and Jie Lu. 2018. Semantic structure-based word embedding by incorporating concept convergence and word divergence. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[22] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao. 2013. Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. 733–738.

[23] Xin Liu, Yong Liu, and Xiaoli Li. 2016. Exploring the context of locations for personalized location recommendations. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 1188–1194.

[24] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, Nov. (2008), 2579–2605.

[25] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2018. A contextual attention recurrent architecture for context-aware venue recommendation. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 555–564.

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*. 3111–3119.

[27] Niluthpol Chowdhury Mithun, Rameswar Panda, Evangelos E. Papalexakis, and Amit K. Roy-Chowdhury. 2018. Webly supervised joint embedding for cross-modal image-text retrieval. In *Proceedings of the 26th ACM International Conference on Multimedia*. 1856–1864.

[28] Anna Potapenko, Artem Popov, and Konstantin Vorontsov. 2017. Interpretable probabilistic embeddings: Bridging the gap between topic models and neural networks. In *Proceedings of the Conference on Artificial Intelligence and Natural Language*. Springer, 167–180.

[29] Tieyun Qian, Bei Liu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2019. Spatiotemporal representation learning for translation-based POI recommendation. *ACM Trans. Inf. Syst.* 37, 2 (2019), 1–24.

[30] Yaqiong Qiao, Xiangyang Luo, Chenliang Li, Hechan Tian, and Jiangtao Ma. 2020. Heterogeneous graph-based joint representation learning for users and POIs in location-based social network. *Inf. Process. Manag.* 57, 2 (2020), 102151.

[31] Hossein A. Rahmani, Mohammad Aliannejadi, Rasoul Mirzaei Zadeh, Mitra Baratchi, Mohsen Afsharchi, and Fabio Crestani. 2019. Category-aware location embedding for point-of-interest recommendation. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*. 173–176.

[32] NCSS Statistical Software. [n.d.]. Paired T-Test for Superiority by a Margin. Retrieved from: NCSS.com.

[33] Yue Wang, Meng Chen, Xiaohui Yu, and Yang Liu. 2017. LCE: A location category embedding model for predicting the category labels of POIs. In *Proceedings of the International Conference on Neural Information Processing*. Springer, 710–720.

[34] Yiling Wu, Shuhui Wang, and Qingming Huang. 2018. Learning semantic structure-preserved embeddings for cross-modal retrieval. In *Proceedings of the 26th ACM International Conference on Multimedia*. 825–833.

[35] 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*. 15–24.

[36] Bo Yan, Krzysztof Janowicz, Gengchen Mai, and Song Gao. 2017. From ITDL to Place2Vec: Reasoning about place type similarity and relatedness by learning embeddings from augmented spatial contexts. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 35.

[37] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. 2019. Revisiting user mobility and social relationships in LBSNs: A hypergraph embedding approach. In *Proceedings of the World Wide Web Conference*. 2147–2157.

[38] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. 2015. NationTelescope: Monitoring and visualizing large-scale collective behavior in LBSNs. *J. Netw. Comput. Applic.* 55 (2015), 170–180.

[39] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Trans. Intell. Syst. Technol.* 7, 3 (2016), 30.

[40] Jing Yang and Carsten Eickhoff. 2018. Unsupervised learning of parsimonious general-purpose embeddings for user and location modeling. *ACM Trans. Inf. Syst.* 36, 3 (2018), 32.

[41] Hongzhi Yin, Bin Cui, Zi Huang, Weiqing Wang, Xian Wu, and Xiaofang Zhou. 2015. Joint modeling of users' interests and mobility patterns for point-of-interest recommendation. In *Proceedings of the 23rd ACM International Conference on Multimedia*. 819–822.

[42] Haochao Ying, Jian Wu, Guandong Xu, Yanchi Liu, Tingting Liang, Xiao Zhang, and Hui Xiong. 2019. Time-aware metric embedding with asymmetric projection for successive POI recommendation. *World Wide Web* 22, 5 (2019), 2209–2224.

[43] Fuqiang Yu, Lizhen Cui, Wei Guo, Xudong Lu, Qingzhong Li, and Hua Lu. 2020. A category-aware deep model for successive POI recommendation on sparse check-in data. In *Proceedings of the Web Conference*. 1264–1274.

[44] Jianglong Zhang, Liqiang Nie, Xiang Wang, Xiangnan He, Xianglin Huang, and Tat Seng Chua. 2016. Shorter-is-better: Venue category estimation from micro-video. In *Proceedings of the 24th ACM International Conference on Multimedia*. 1415–1424.

[45] Jia-Dong Zhang and Chi-Yin Chow. 2015. GeoSoCa: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 443–452.

[46] Shenglin Zhao, Tong Zhao, Irwin King, and Michael R. Lyu. 2017. Geo-Teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web*. 153–162.

[47] Wayne Xin Zhao, Feifan Fan, Ji-Rong Wen, and Edward Y. Chang. 2018. Joint representation learning for location-based social networks with multi-grained sequential contexts. *ACM Trans. Knowl. Discov. Data* 12, 2 (2018), 1–21.

[48] Wayne Xin Zhao, Ningnan Zhou, Aixin Sun, Ji-Rong Wen, Jialong Han, and Edward Y. Chang. 2018. A time-aware trajectory embedding model for next-location recommendation. *Knowl. Inf. Syst.* 56, 3 (2018), 559–579.

[49] Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. 2020. Hierarchy-aware global model for hierarchical text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1106–1117.

[50] Ningnan Zhou, Wayne Xin Zhao, Xiao Zhang, Ji-Rong Wen, and Shan Wang. 2016. A general multi-context embedding model for mining human trajectory data. *IEEE Trans. Knowl. Data Eng.* 28, 8 (2016), 1945–1958.

[51] Yang Zhou and Yan Huang. 2018. DeepMove: Learning place representations through large scale movement data. In *Proceedings of the IEEE International Conference on Big Data (Big Data)*. IEEE, 2403–2412.

Received September 2020; revised June 2021; accepted July 2021