TME: Tree-guided Multi-task Embedding Learning towards Semantic Venue Annotation

RONGHUI XU, School of Software, Shandong University, China
MENG CHEN*, School of Software, Shandong University, China
YONGSHUN GONG, 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
LIQIANG NIE, Harbin Institute of Technology (Shenzhen), China

The prevalence of location-based services has generated a deluge of check-ins, enabling the task of human mobility understanding. Among the various types of information associated with the check-in venues, categories (e.g., *Bar* and *Museum*) are vital to the task, as they often serve as excellent semantic characterization of the venues. Despite its significance and importance, a large portion of venues in the check-in services do not have even a single category label, such as up to 30% of venues in the Foursquare system lacking category labels. We therefore address the problem of semantic venue annotation, i.e., labeling the venue with a semantic category. Existing methods either fail to fully exploit the contextual information in the check-in sequences, or do not consider the semantic correlations across related categories. As such, we devise a Tree-guided Multi-task Embedding model (TME for short) to learn effective representations of venues and categories for the semantic annotation. TME jointly learns a common feature space by modeling multi-contexts of check-ins and utilizes the predefined category hierarchy to regularize the relatedness among categories. We evaluate TME over the task of semantic venue annotation on two check-in datasets. Experimental results show the superiority of TME over several state-of-the-art baselines.

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

Additional Key Words and Phrases: Semantic venue annotation, Human mobility, Check-in analysis, Embedding learning

1 INTRODUCTION

With the rapid development of location-based social networks (LBSNs), it becomes more important to understand the venues we visit. LBSNs usually utilize geographical information (e.g., latitude and longitude) or street address to represent a venue. Recently, some LBSNs such as Foursquare enable users to explicitly indicate the semantic categories (e.g., *Noodle House* and *Museum*) of venues. These semantic categories are utilized to complement the representation of a venue, which are crucial for assisting users in exploring new venues as well as providing

*Corresponding author.

Authors' addresses: Ronghui Xu, School of Software, Shandong University, Jinan, China, xrhics@163.com; Meng Chen, School of Software, Shandong University, Jinan, China, mchen@sdu.edu.cn; Yongshun Gong, School of Software, Shandong University, Jinan, China, ysgong@sdu.edu.cn; Yang Liu, Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo, Canada, yangliu@wlu.ca; Xiaohui Yu, School of Information Technology, York University, Toronto, Canada, xhyu@yorku.ca; Liqiang Nie, Harbin Institute of Technology (Shenzhen), Shenzhen, China, nieliqiang@gmail.com.

1046-8188/2023/2-ART $15.00
https://doi.org/10.1145/3582553

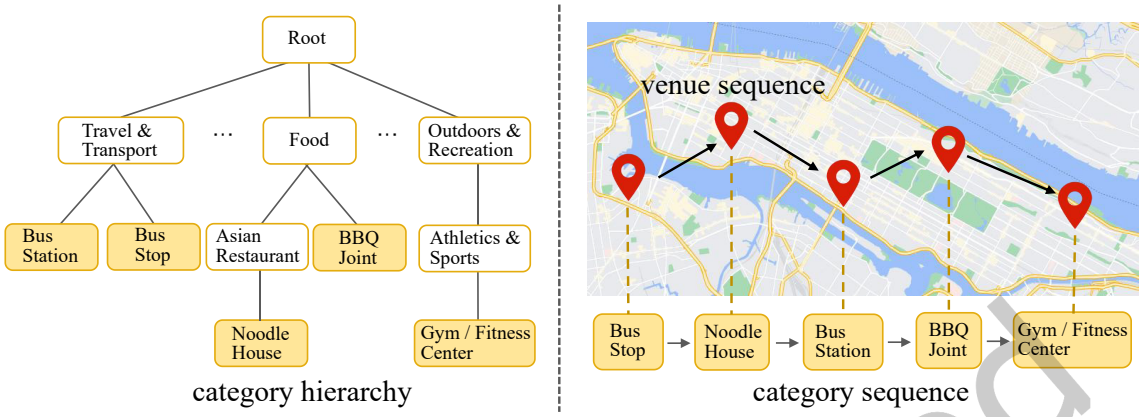


Fig. 1. Multi-view information embodied in check-ins.

semantic-based recommendation services [12, 46]. Despite their benefits, existing studies [16, 20, 41] observe that about 30% of venues lack semantic categories in Foursquare. Therefore, semantic venue annotation, namely automatically and precisely annotating a venue with the most-likely category from all categories, is highly desirable and has gradually become a research hotspot of urban computing [8, 35, 45].

To tackle this problem, existing methods usually employ human mobility data (e.g., check-ins of a user over a period of time) for semantic venue annotation. A common methodology is to construct descriptive features (e.g., visiting frequency and distribution of visiting time) regarding individual venues based on users' check-ins and build a classification model to infer venue categories [5, 13, 18, 41], which suffers from several major problems. Firstly, the relations between venues and their contexts in check-in sequences are overlooked. Such information could help to construct discriminative venue features by embedding the semantic properties of check-ins into low-dimensional dense vectors. Secondly, the semantic correlations across venue categories are largely untapped. As shown in the left part of Fig. 1, semantic categories form a tree structure, whereby the category nodes are not independent but hierarchically correlated. How to model these inter-related labels to boost the discriminative feature engineering is worthy of investigating.

Considering the complex relationships of venues and categories, we try to explore the use of representation learning for the task of semantic venue annotation via check-ins. Recently, representation learning has been shown to be highly effective for a related but different task of learning venue semantics based on check-in sequences (without category information) [9, 31, 42, 48]. They learn venue representations by predicting the co-occurrence of venues in the given ***venue context*** (i.e., venues that immediately precede and follow the given venue within a window in a venue sequence). When category tags for the venues in a check-in sequence are considered, a category sequence is formed (see the right part of Fig. 1). However, we often find that two very different venue sequences may correspond to the same or similar category sequence, as many venues share a common category tag. Such category sequence information imposes additional constraints on the representation of venues, and if properly utilized, it could improve representation quality. It is therefore desirable to additionally model the semantic relations encoded in the ***category context*** (which categories those venues in the venue context belong to) from the category sequences.

In this paper, we present a Tree-guided Multi-task Embedding model, abbreviated as TME, for semantic venue annotation. TME is able to simultaneously project venues and categories in a latent space, considering users' sequential behaviors on venues and venue categories, and the hierarchical structure of categories. Specifically,

TME consists of two components: the sequential and categorical embedding component, and the tree-guided multi-task learning component. The main goal of the first component is to learn the representations of venues and categories using the venue sequences and the corresponding category sequences. Here, we separately build the venue-venue co-occurrences and the venue-category co-occurrences by modeling the venue context and the category context, and factorize them to learn these representations. In this part, both sequential patterns and categorical characteristics included in the two kinds of contexts can help us to deliver better representations for venues and categories.

The second component aims to enrich and enhance these representations by modeling the known venue categories and the predefined category hierarchy. To model relations between venues and their categories, we factorize the venue-category label matrix into the product of venue representations and the corresponding category ones. Further, as the categories are hierarchically correlated, we turn to leverage a tree-guided multi-task learning algorithm to capture the inter-task relatedness (i.e., similarities among inter-related categories determined by the category tree in our study) for learning more discriminative representations of venues and categories. The two components are accomplished within a unified framework via these representations, and distributional venue semantics are regulated accordingly. Finally, we perform parameter inference using the alternating optimization strategy.

The contributions can be summarized as follows:

- We propose a Tree-guided Multi-task Embedding model for semantic venue annotation. To the best of our knowledge, TME is the first work that simultaneously models multiple kinds of sequential patterns from check-in sequences and the semantic correlation across related categories from the category hierarchy to solve the problem of semantic venue annotation.
- We jointly model the category context (which is formed by the categories of adjacent venues in the check-in sequence and thus reflects aggregated semantics) and the venue context (which embodies the sequential relationships between individual venues) from check-in sequences to learn venue semantics. Moreover, we encode the category relatedness constrained by a predefined hierarchical structure in the representations of venues and categories.
- We justify our TME model on two check-in datasets collected from Foursquare and evaluate its performance via the task of semantic venue annotation. TME demonstrates significant performance gains over these baseline methods according to the paired t-test. We have put up a shared GitHub folder and released our codes of TME to facilitate the research communities[1].

The remainder of this paper is structured as follows: Section 2 introduces related methods on semantic venue annotation, check-in embedding learning, hierarchy embedding learning, as well as recent trends on multi-task learning. Section 3 defines the preliminary concepts used in this work. Section 4 details our proposed TME method. Section 5 provides the experimental results and analyses, followed by the conclusion in Section 6.

2 RELATED WORK

Our work involves studies on semantic venue annotation, check-in embedding learning, hierarchy embedding learning, and multi-task learning.

2.1 Semantic Venue Annotation

There exist two kinds of methods for semantic venue annotation: personalized and non-personalized. The former models individual differences and labels different semantics for a venue. Some methods [8, 27, 35, 51] leverage the smartphone log data or GPS trajectories to infer personalized venue semantics. Specifically, they either employ embedding methods to learn effective representations of venues or construct features using heuristics from

[1]https://github.com/xrhics/TME

multi-context information to recognize personalized semantics. On the contrary, non-personalized semantic annotation focuses only on the common semantic representation of venues. Existing methods [13, 29, 46] have demonstrated that various features extracted from user generated contents are effective to depict non-personalized venue semantics. For example, Zhang et al. [46] extract features from multi-modal data including visual data, acoustic data and textual data, and label the bite-sized video clips with venue categories. Meng et al. [29] propose a feature-level fusion method based on the text-image pairs collected from social networks to label venues with semantics.

In addition, some studies model users' check-ins to make non-personalized semantic venue annotation, which is the most similar to our work. On one hand, some methods [5, 20, 41] design features manually and train multi-label classifiers to recognize venue semantics. For instance, Ye et al. [41] study temporal features and human mobility features (e.g., the total number of check-ins) to train a SVM for category classification. Li et al. [20] extract the similar user pattern by capturing the similarities among different users' check-in activities, and train a multi-label classifier for annotating semantic tags of venues. On the other hand, some work [16, 32, 34] learn representations of venues from check-ins and make semantic annotation accordingly. For example, based on check-ins and rating behaviors of users, He at al. [16] propose a spatial-temporal model that labels every venue with semantic and emotional tags simultaneously. Wang et al. [34] learn the embeddings of venues and categories by considering both the sequential contexts and the correlation between venues and their categories. Rahmani et al. [32] present CATAPE which models both the geographical influence of venues and the category-related labels of venues to generate venue embeddings. However, the above methods either fail to fully model the contextual information (including both the venue context and the category context) in check-in sequences, or do not consider the semantic correlation (i.e., the hierarchical structure) across related categories.

2.2 Check-in Embedding Learning

Nowadays, it has become convenient to capture users' mobility patterns via check-ins in LBSNs. Such check-in data can be used to enhance the capabilities of various location-based applications, such as point-of-interest (POI) recommendation and user profiling. It hence has aroused the interest of the trajectory mining community. Many studies [47, 50] learn venue embeddings with check-in data based on the popular word embedding models (e.g., Word2vec [30]). For example, Zhou et al. [50] construct a multi-context trajectory embedding model called MC-TEM to learn trajectory embeddings, which takes users, surrounding POIs, the corresponding category labels and the temporal factor as the context to predict the target POI. Zhao et al. [47] propose an embedding model (Geo-Teaser) that mainly captures the geographic information and the temporal features inherent in the check-in trajectories. However, these methods either learn venue embeddings or category embeddings, instead of modeling the relations between venues and categories, which cannot be applied to the task of semantic venue annotation directly.

Further, there are some studies that improve the quality of check-in embeddings by considering external information (e.g., social relations, taxi trip data, and tag words of venues) [1, 38]. For example, Aliannejadi et al. [1] leverage users' ratings on venues and learn the latent vectors of users and venues based on matrix factorization. Yang et al. [38] consider both the social relations of users and the user mobility to construct the LBSN hypergraph, and present a model named LBSN2Vec to learn node representations. Unfortunately, these embedding methods require specific check-in data with additional information (e.g., social relations, venue tags).

In addition, deep learning models have gained a breakthrough in mobility data mining recently. Luca et al. [25] survey recent studies and propose a perspective on the leading deep learning solutions (including fully connected networks, recurrent neural networks, attention mechanisms, and convolutional neural networks) to multiple mobility tasks. In particular, some methods [28, 37, 43] model the check-in sequences based on recurrent neural networks (RNNs) and generate the venue embeddings as by-products. For example, Yang et al. [37] directly utilize

a framework of RNN to model the sequential relatedness of venues from trajectories. Manotumruksa et al. [28] leverage the 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] consider both the sequential and categorical information of venues and propose a deep model named CatDM based on LSTM encoders and decoders for next venue prediction. However, these methods focus on modeling the sequential patterns of check-in sequences without considering the hierarchical structure of venue categories.

2.3 Hierarchy Embedding Learning

In this study, we model the predefined hierarchical structure of venue categories to enhance the representations of venues and categories, which is related to researches on hierarchy embedding learning. Some methods [2, 23, 49] model the hierarchy-based relations between words and learn distributed representations for words. For example, Liu et al. [23] leverage the structure of WordNet and incorporate the constraints of concept convergence and word divergence into the word2vec model to generate the semantic structure-based word embeddings, in which they assume that a word tends to be away from those words at the same level and close to the center of words on the lower level in the embedding space. Alsuhaibani et al. [2] present a hierarchical word embedding method which models the direct hypernymy relations between the hypernym and hyponym words as well as the indirect and the full hierarchical hypernym path.

Moreover, there exist some methods [7, 15, 33] that aim to predict the categories of products in the e-commerce system, defined as the hierarchical classification task because categories in most e-commerce websites are organized as a hierarchical tree. Gao et al. [15] propose a deep hierarchical classification framework to solve the two problems (hierarchical representation and hierarchical inconsistency) in hierarchical category classification. Chen et al. [7] propose a neural product categorization model to predict the fine-grained categories for products, considering both the product content and a predefined product category vocabulary. Tan et al. [33] present a novel paradigm for product categorization based on machine translation. Specifically, they translate the text descriptions of a product into a sequence of categories representing a root-to-leaf path in the hierarchical tree of categories. In addition, Yan et al. [36] consider the prior category graphs and develop two semantics-preserving graph propagation modules to enhance both category and region representations for zero-shot object detection. In contrast to the methods mentioned above that mainly model the relations between nodes in the hierarchical structure, the proposed TME models two kinds of linear context (the venue context and the category context) and the hierarchical context (from category hierarchy) collaboratively.

Chen et al. [11] propose a pre-trained venue category embedding model named Hier-CEM, which embeds the hierarchical structure of categories and utilizes multiple types of context to generate a latent representation for each venue category. Though both the proposed TME and Hier-CEM model the hierarchical structure of categories and the check-in sequences, they are different in the following aspects. First, the purpose of the two methods is different: Hier-CEM focuses on generating the pre-trained category embeddings which can be used in many tasks such as venue semantic study and next category prediction; the proposed TME generates both venue embeddings and category embeddings and is designed for semantic venue annotation. As Hier-CEM does not generate venue embeddings, it cannot be utilized to make semantic venue annotation directly. Second, the idea of mining sequential patterns from check-in sequences is different: Hier-CEM merely models the category context from the check-in sequences; the proposed TME collaboratively models the venue context and the category context, and makes direct connections between venue embeddings and category embeddings accordingly. Third, the method of mining the hierarchical structure of categories is different: Hier-CEM separately constructs two kinds of hierarchical context using the category hierarchy and establishes connections between categories in the check-in sequence and categories in the hierarchy; the proposed TME introduces a tree-guided multi-task learning method to leverage the hierarchical relations among categories. Finally, the application of venue label

Table 1. Notations and Descriptions.

Notations	Descriptions
l, e, c	Target venue, Context venue, Category
$\mathcal{L}, C, \mathcal{T}$	Venue set, Category set, Trajectory set
N_l, N_c	Number of venues and categories
D	Dimensionality of embedding space
$\mathbf{M}^l \in \mathbb{R}^{N_l \times N_l}$	Venue co-occurrence positive PMI matrix
$\mathbf{M}^c \in \mathbb{R}^{N_l \times N_c}$	Venue category positive PMI matrix
$\mathbf{Y} \in \mathbb{R}^{N_l \times N_c}$	Venue category label matrix
$\mathbf{Q} \in \mathbb{R}^{D \times D}$	Coefficient matrix
$\mathbf{L} \in \mathbb{R}^{N_l \times D}$	Target venue embedding matrix
$\mathbf{E} \in \mathbb{R}^{N_l \times D}$	Context venue embedding matrix
$\mathbf{C} \in \mathbb{R}^{N_c \times D}$	Category embedding matrix

information (i.e., categories) is different: Hier-CEM does not consider the label information; the proposed TME captures the relations between venues and their corresponding categories by decomposing the venue-category label matrix into the product of venue embeddings and category embeddings.

2.4 Multi-task Learning

semantic venue annotation exhibits dual heterogeneities, i.e., a single annotation task leverages features of venues encoded in a latent space, and multiple annotation tasks are related to each other according to the hierarchical relations of categories. Thus we introduce the studies on multi-task learning briefly.

Regarding the inter-related learning tasks, multi-task learning can strengthen the generalization ability of a learning task or separability of object classes in the form of a learning paradigm [4]. Multi-task learning needs to consider the association among different tasks in order to share information properly. Taking into account the differences among these tasks, some methods [6, 21, 26] optimize both the multi-task function and the inter-task relationships. For instance, Ling et al. [21] propose CTNet to predict person identity and attributes at the same time by integrating four different tasks into a multi-task network. Luo et al. [26] present a semi-supervised feature analyzing framework, which integrates the adaptive optimal similarity matrix learning into the procedure of feature selection. Cheikhrouhou et al. [6] customize a multi-task learning framework to train each portion of architecture by back-propagating a common combined loss for all the tasks.

In addition, there also exist some multi-task learning methods [14, 24] where the tasks are organized with a hierarchical structure. These methods utilize regularization terms to couple inter-related tasks according to the tree structure. Using a tree guided sparse group lasso regularization term, Lu et al. [24] model the hierarchical structure of traffic sign recognition tasks and learn classifiers across tasks jointly. Fan et al. [14] utilize a manifold regularization term to ensure the model parameters of similar node classifiers sharing some components and having stronger correlations.

3 PROBLEM DEFINITION

We first define the preliminary concepts and the problem statement used in this study, and then list the notations and their descriptions in Table 1.

Definition 1 (Venue category hierarchy). *Without loss of generality, we adopt the categories used in Foursquare, which constitute a five-layer hierarchical structure, as the category hierarchy.*

The entire tree can be viewed here[2]. The top layer includes 10 categories, and the average number of children of a non-leaf category is 12.87.

Definition 2 (Check-in). *A check-in represents that a user u visits a venue l at time t. We denote it with a tuple $< u, l, c, t >$, where c demonstrates the category of the visited venue, which could be at any layer of the category hierarchy.*

Note that a venue is labeled with at most a category tag, and some may not be labeled with any category. For the latter, we assign the "NULL" tag to them.

Definition 3 (Trajectory). *We sort a user's check-ins according to the visited time, and obtain a check-in trajectory.*

Problem statement. Given users' *check-in trajectories* and *venue category hierarchy*, our goal is to project both venues and categories in a latent semantic space and predict the category label (which could be at any layer of the category hierarchy) for those venues with the "NULL" tag.

4 METHODOLOGY

We first introduce the framework of the proposed TME, and then detail the two components.

4.1 Overview of TME

Users' check-ins usually contain multi-view information, e.g., the venue sequences and the category sequences imply human mobility patterns, the categorical information of venues indicate aggregated semantics, and the category hierarchy encodes the relatedness among categories (cf. Fig. 1). We thus fully utilize the information and propose a Tree-guided Multi-task Embedding model for semantic venue annotation. TME consists of the sequential and categorical embedding component and the tree-guided multi-task learning component, which is built based on the following two basic assumptions:

- **Sequential and categorical embedding component**: We assume that there exists a common feature space for venues and categories. In the embedding space, venues appearing with similar venue context or category context tend to have similar semantics and ought to be mapped closer.
- **Tree-guided multi-task learning component**: We assume that the category tree structure encodes the similarities among inter-related categories and multiple annotation tasks are related to each other accordingly. Leveraging such knowledge could help to learn more discriminative representations of venues and categories.

Fig. 2 shows the framework of the proposed TME. Here we illustrate TME with a running example. Let us suppose that a user has a trajectory l_1(*Bus Stop*) $\rightarrow$ l_2(*Noodle House*) $\rightarrow$ l_3(*Bus Station*) $\rightarrow$ l_4(*NULL*) $\rightarrow$ l_5(*Gym/Fitness Center*). In the component of sequential and categorical embedding, we first fetch the venue sequences and the corresponding category sequences from the check-in trajectories, and model the relations between the target venue and its two kinds of sequential context. Specifically, we construct the venue co-occurrence positive point-wise mutual information (PMI) matrix and the venue category positive PMI matrix by modeling the venue context and the category context separately. Then we factorize them to learn venue embeddings and category embeddings, expecting that venues (e.g., l_1 and l_3) with similar context are close to each other in the feature space.

[2]https://developer.foursquare.com/docs/categories

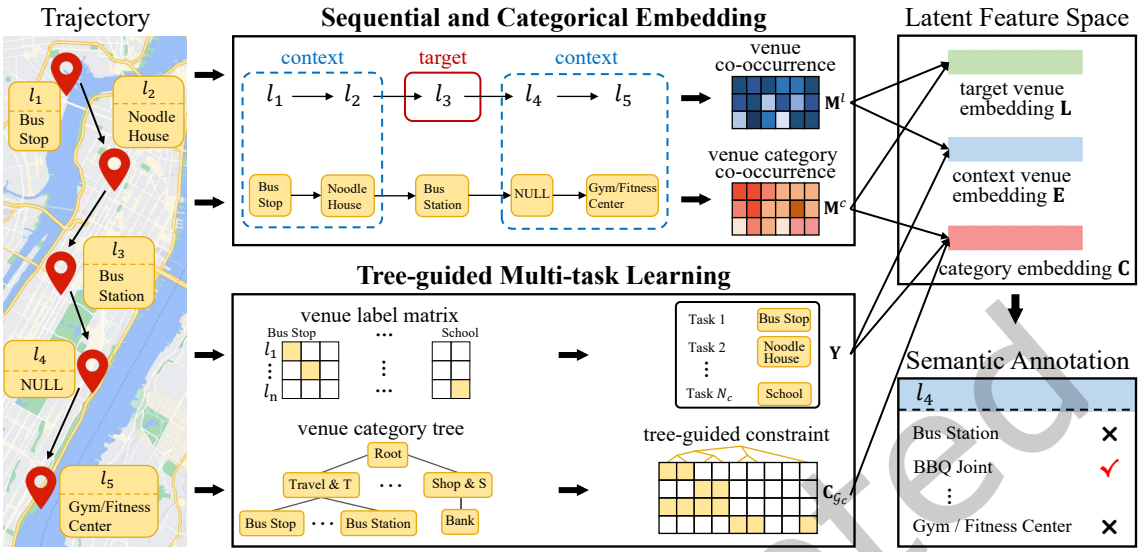


Fig. 2. Overview of the proposed TME framework. TME takes the trajectories (including venues and their categories) as inputs. The sequential context information is modeled in the sequential and categorical embedding component; the label information (i.e., categories) of venues and the category structure are embedded in the tree-guided multi-task learning component. Two components are optimized jointly to generate the feature representations of venues and categories, followed by semantic venue annotation based on these representations.

In the component of tree-guided multi-task learning, we model both the label information (i.e., categories) of venues and the category hierarchical structure to better learn discriminative feature representations. Namely, we build the venue category label matrix $\mathbf{Y} \in \mathbb{R}^{N_l \times N_c}$ (where N_l and N_c are the number of venues and categories respectively) and factorize it into the product of context venue embeddings and category embeddings, expecting that it has a large probability that a venue (e.g., l_3) is labeled as its category (e.g, *Bus Station*) based on the corresponding venue vector and category vector. Meanwhile, we explicitly model the relatedness among these annotation tasks using the predefined category hierarchy. For example, some venues are annotated as *Bus Station* and some are annotated as *Bus Stop*, and the categories *Bus Station* and *Bus Stop* have the same father node in the category tree. Thus we add this tree-guided constraint to refine these category embeddings accordingly. Note that these venue embeddings and category embeddings are regulated by the two components. Finally, we utilize the venue embeddings and the category embeddings to make semantic venue annotation.

4.2 Sequential and Categorical Embedding

We propose to learn the embeddings of venues from its associated context venues and categories based on the check-in trajectories. First, we model the sequential patterns and follow the distributional hypothesis that venues occurring in similar contexts tend to have similar semantics and should appear closer in the latent embedding space. Specifically, given a target venue in the check-in trajectory, we define its context as the venues that immediately precede and follow the target venue within a small window in the trajectory. We adopt the Skip-gram model, which leverages a target venue to predict context venues, to generate venue embeddings. Since the objective of the Skip-gram model has been proven to be equivalent to implicitly factorizing a shifted positive co-occurrence PMI matrix [10, 19], we first compute the $N_l \times N_l$ PMI matrix based on the check-in trajectory set

$\mathcal{T}$. The PMI value in the $\langle l, e\rangle$ entry is computed as

$$\text{PMI}(\mathcal{T})_{l,e} = \log \frac{\sharp(l,e) \times |\mathcal{T}|}{\sharp(l) \times \sharp(e)}. \tag{1}$$

We use $\sharp(l, e)$ to denote the number of times that venues l and e co-occur in $\mathcal{T}$, where a venue co-occurrence happens when a venue e lies in the context of l. Similarly, $\sharp(l)$ and $\sharp(e)$ are the number of times that l and e occur in $\mathcal{T}$ independently. $|\mathcal{T}|$ is the total number of venue-context pairs in the set of trajectories. Note that the context window size affects the PMI values, which will be evaluated in the experiments.

Further, we learn venue embeddings by decomposing the PMI matrix. That is,

$$v_l^T v'_e \approx \text{PMI}(\mathcal{T})_{l,e}, \tag{2}$$

where v_l is the embedding of the target venue l and v'_e is the embedding of the context venue e. Here we distinguish between the roles of target venues and context venues following the Word2vec model and learn two embedding vectors (i.e., target venue embedding and context venue embedding) per venue.

Sometimes, modeling the venue context to learn venue semantics is not enough. For example, two different venue sequences may correspond to the same category sequence, e.g., *Noodle House* → *Bus Station* → *Gym / Fitness Center*, as many venues are associated with the same semantic category. Such category sequence information imposes additional constraints on the venue embeddings. Therefore, we additionally model the category context of venues to enhance these embeddings. Specifically, given a venue l in a trajectory, we consider the categories of venues visited before and after l based on a predefined window size as the category context. Similarly, we compute the frequency of venue and category co-occurrence from the trajectory set, and decompose PMIs of venue and category pairs to learn both venue and category embeddings,

$$\begin{aligned} \text{PMI}(\mathcal{T})_{l,c} &= \log \frac{\sharp(l,c) \times |\mathcal{T}|}{\sharp(l) \times \sharp(c)}, \\ v_l^T v'_c &\approx \text{PMI}(\mathcal{T})_{l,c}, \end{aligned} \tag{3}$$

where v'_c is the embedding of the context category c, $\sharp(l, c)$ denotes the frequency that venue l and category c co-occur, and other terms have the same definitions as those in Equation (1).

Considering both the sequential and categorical patterns in learning venue embeddings, we minimize the following objective function over all the check-in trajectories,

$$\min_{\mathbf{L},\mathbf{E},\mathbf{C}} ||\mathbf{M}^l - \mathbf{L}\mathbf{E}^{\mathrm{T}}||_F^2 + ||\mathbf{M}^c - \mathbf{L}\mathbf{C}^{\mathrm{T}}||_F^2, \tag{4}$$

where $\mathbf{M}^l$ is the $N_l \times N_l$ matrix of positive PMI values which measures each co-occurrence of venues, i.e., $\mathbf{M}^l(l, e) = \max(0, PMI(\mathcal{T})_{l,e})$, and $\mathbf{M}^c$ denotes the $N_l \times N_c$ matrix of positive PMI values which measures every co-occurrence of venues and categories, i.e., $\mathbf{M}^c(l, c) = \max(0, PMI(\mathcal{T})_{l,c})$. $\mathbf{L}$ is the $N_l \times D$ matrix of target venue embeddings, $\mathbf{E}$ is the $N_l \times D$ matrix of context venue embeddings, and $\mathbf{C}$ is the $N_c \times D$ matrix of category embeddings, where D is the size of the latent embedding space. We jointly factorize the positive PMI matrices $\mathbf{M}^l$ and $\mathbf{M}^c$ to learn context/target venue embeddings and category embeddings by minimizing the square errors for all co-occurrences, where F indicates the Frobenius norm. As such, the contexts (including venues and categories) with the same target venue are squeezed into a corner of the latent semantic space.

4.3 Tree-guided Multi-task Learning

Categories as the label information of venues are not independent but hierarchically correlated, due to the tree structure of categories defined by experts. In other words, there is a certain correlation between the tasks of venue annotation associated with parent and sub-category. Therefore, a tree-guided multi-task learning method

is developed to leverage the hierarchical relations among categories to learn more discriminative representations of venues and categories.

We first utilize the label information of venues and build the venue category label matrix $\mathbf{Y} \in \mathbb{R}^{N_l \times N_c}$. Each row of $\mathbf{Y}$ is a one-hot vector for representing the category label of a venue. As context venues and context categories have one-to-one correspondence and context embeddings are usually used as latent features, we decompose $\mathbf{Y}$ into the product of venue embeddings $\mathbf{E}$ and category embeddings $\mathbf{C}$. Meanwhile, we treat each category as a task and regularize the relatedness among tasks. Specifically, as shown in the category hierarchy of Fig. 1, for each category c (e.g., *Travel & Transport*) in the tree, we collect all its children nodes (e.g., *Bus Station* and *Bus Stop*) and construct a group $\mathcal{G}_c$ containing c and its children nodes, whereby tasks (categories) within the same group bear large semantic relatedness. In this case, a subset of highly correlated tasks may share a common set of relevant features, whereas weakly related tasks are less likely to be affected by the same features. Thus, we assign a weight e_c to each category group to capture the strength of relatedness among tasks within the same group $\mathcal{G}_c$ and define the new loss function as follows,

$$\min_{\mathbf{C},\mathbf{E}} ||\mathbf{Y} - \mathbf{E}\mathbf{C}^{\mathrm{T}}||_F^2 + \lambda_1 \sum_{c \in C} e_c ||\mathbf{C}_{\mathcal{G}_c}||_{2,1}, \tag{5}$$

where $\mathbf{C}_{\mathcal{G}_c} = \{\mathbf{C}_i : c_i \in \mathcal{G}_c\} \in \mathbb{R}^{|\mathcal{G}_c| \times D}$ is the embedding matrix of categories from group $\mathcal{G}_c$ and D is the latent feature dimension. $||\mathbf{C}_{\mathcal{G}_c}||_{2,1} = \sum_{d=1}^{D} \sqrt{\sum_{c_i \in \mathcal{G}_c} (\mathbf{C}_{id})^2}$ is the $\ell_{2,1}$-norm regularization (i.e., group lasso [44]). Lasso uses ℓ_1 norm for regularization to obtain a sparse solution; group lasso groups all variables and then penalizes the ℓ_2 norm of each group in the objective function, which is imposed to select discriminative features for each classification task. In this way, we can simultaneously learn task-sharing and task-specific features. λ_1 is the weight of the regularization term.

Further, a category group with large semantic similarity among categories indicates the classification tasks are highly correlated and have a set of task-sharing features. Such features are important for improving classification performance. Thus we expect that a group $\mathcal{G}_c$ containing semantically similar categories tend to have a large weight e_c. The large weight is critical for feature selection, which only selects a few highly discriminative features and results in sparse feature representations for categories in the group. These sparse discriminative features make it easier to linearly separate the correlated tasks. Specifically, we model the semantic relatedness among tasks based on the feature space and use the affinity measurement of the node group following [22] to calculate e_c. We first calculate the pairwise cosine similarity S_{ij} between two categories c_i and c_j based on the category embeddings, $S_{ij} = \mathbf{C}_i \cdot \mathbf{C}_j / ||\mathbf{C}_i|| \cdot ||\mathbf{C}_j||$. Considering that the number of categories in each group is different, we define a scaled vector $\mathbf{A}_c \in \mathbb{R}^{N_c}$ for each category c,

$$\mathbf{A}_{ct} = \begin{cases} \frac{1}{\sqrt{|\mathcal{G}_c|}}, & if\ t \in \mathcal{G}_c \\ 0, & otherwise. \end{cases} \tag{6}$$

We then calculate e_c considering both the similarity matrix $\mathbf{S}$ and the scaled vector $\mathbf{A}_c$,

$$e_c = \mathbf{A}_c^T \mathbf{S} \mathbf{A}_c. \tag{7}$$

Finally, we integrate the two components (Equations (4) and (5)) and obtain the objective function,

$$\begin{aligned} \min_{\mathbf{L},\mathbf{E},\mathbf{C}} \Gamma = & ||\mathbf{M}^l - \mathbf{L}\mathbf{E}^{\mathrm{T}}||_F^2 + ||\mathbf{M}^c - \mathbf{L}\mathbf{C}^{\mathrm{T}}||_F^2 \\ & + ||\mathbf{Y} - \mathbf{E}\mathbf{C}^{\mathrm{T}}||_F^2 + \lambda_1 \sum_{c \in C} e_c ||\mathbf{C}_{\mathcal{G}_c}||_{2,1} \\ & + \lambda(||\mathbf{L}||_F^2 + ||\mathbf{E}||_F^2 + ||\mathbf{C}||_F^2), \end{aligned} \tag{8}$$

where λ is the regularization parameter.

Table 2. Check-in Data Statistics.

	♯user	♯venue	♯category	♯check-in
TKY	9,548	12,605	103	1,270,977
NYC	11,097	15,632	138	799,825

To learn the parameters including target venue embeddings **L**, context venue embeddings **E**, and category embeddings **C** of TME, we employ the alternating optimization strategy and detail the process in the appendix.

5 EXPERIMENTS

We begin by introducing the datasets and experimental settings; then we report the results using both quantitative and qualitative analyses. Through these experiments, we solve the following Research Questions (RQ):

- **RQ1**: Could the proposed TME improve the performance of semantic venue annotation?
- **RQ2**: Could we enhance the effectiveness of check-in embedding methods by modeling the category context which reflects aggregated semantics?
- **RQ3**: Does the proposed TME benefit from the component of tree-guided multi-task learning?

5.1 Datasets and Settings

Check-in data. The Foursquare data includes check-ins of 18 months collected from Tokyo and New York [39]. Each check-in contains *user ID*, *venue ID*, *venue category* and *timestamp*. To make the proposed model robust, we filter those users and venues with less than 20 check-ins. After pre-processing, the data statistics of TKY (short for Tokyo) and NYC (short for New York) datasets are shown in Table 2, where ♯user, ♯venue, ♯category, and ♯check-in are the number of users, venues, categories, and check-ins, respectively. Finally, we sort all check-ins of each user according to timestamps. As a user has limited check-ins in a day, we generate a single check-in trajectory for every user using all his/her check-ins.

Venue category hierarchy. We adopt the five-layer hierarchical structure of categories from Foursquare as the category hierarchy. The categories of venues in check-ins could be at any layer of the category hierarchy. The proposed TME only uses the category hierarchy to construct the category group (i.e., a category and its children), so it is applicable to other tree structures, e.g., the three-layer tree structure of Baidu Map.

Experimental settings. For the regularization parameter, we set the default values at $\lambda = 0.1$. We evaluate the effect of model parameters (i.e., context window size, embedding size D and weight λ_1), and employ grid search to select the optimal parameters with small but adaptive step size. All the experiments run on a 1.8GHz Intel Core i7 PC with 16GB main memory.

Evaluation metrics. We mimic the task of semantic venue annotation, i.e., labeling a given venue with the most possible category based on the venue representations and the category ones. In the training stage, we randomly mark off x% (x% = 10%, 20%, 30%) of all the venues and replace their categories with the "NULL" tag. We use these marked off venues and their categories as testing data, and then in the testing stage, we predict their categories.

Given a venue and the category set, we calculate the pairwise similarity between the venue and a candidate category based on cosine similarity of their embeddings, and generate the ranking of categories based on the similarity scores. Finally, we employ four metrics, namely, Macro-F1, Micro-F1, Accuracy and MRR (Mean Reciprocal Rank), to thoroughly measure the performance. For the *Accuracy@k*, given a test venue, if the ground

truth label is within the top k of the prediction set, then a score of 1 is awarded, and else 0. MRR is defined as

$$MRR = \frac{1}{|\mathcal{L}_{test}|} \sum_{i=1}^{|\mathcal{L}_{test}|} \frac{1}{rank_i}, \tag{9}$$

where $\mathcal{L}_{test}$ is the set of test venues, and $rank_i$ is the rank of the true category of venue l_i in our rankings.

5.2 Baselines

We carry out experiments to compare the performance of TME with several state-of-the-art methods:

- **Random**: It randomly chooses a category as the label of a test venue.
- **Majority**: It labels the test venue with the most major category.
- **EP** [41]: This is a traditional SVM classifier, where it extracts explicit patterns (EP) for each venue to obtain aggregated user behaviors as features.
- **STES** [40]: This is a spatio-temporal embedding algorithm. As we do not include check-in timestamps in our model, we adapt STES by considering the venue ID as the feature word, and learn venue representations with the venue sequences.
- **CARA** [28]: This is a deep learning method which leverages RNNs to model both sequence of check-ins and the contextual information (e.g., reviews and time). As in our evaluation there is no contextual information on venues, we adapt CARA and apply a LSTM encoder to model the venue sequences.
- **LCE** [34]: This is a location category embedding model, where it models the relationships between venues and their contexts, and minimizes the distances between venues and their categories.
- **MC-TEM** [50]: This is a multi-context trajectory embedding model, where it considers a venue and its corresponding category as the context and uses the CBOW model to predict the target venue.
- **CATAPE** [32]: This is a two-phase category-aware POI embedding model, consisting of a check-in embedding module and a category embedding module, where it predicts the context categories given both the target category and the target venue.
- **TME-1**: It is a variant of the proposed TME, which removes the tree-guided multi-task learning component from TME and only models the venue context and the category context from check-in trajectories to learn venue and category embeddings.
- **TME-2**: It is another variant of TME, which sets λ_1 at 0 and does not capture the category relatedness constrained by a predefined hierarchical structure.

Among these baselines, Random and Majority are two naive methods for venue annotation. EP is a traditional SVM classifier for semantic venue annotation, which constructs features manually from check-ins. STES and CARA learn venue representations via modeling the check-in sequences. We then take these venue representations as features, and adopt a SVM for semantic annotation. LCE, MC-TEM, and CATAPE are embedding-related methods, which model the local contexts (e.g., surrounding venues, categories) and learn representations of venues and categories.

5.3 Evaluation on Semantic Venue Annotation

5.3.1 Comparison with baselines. We evaluate all the methods on the same datasets and perform 10 runs to obtain 10 values for Macro-F1 and Micro-F1, and calculate the mean of 10 values. We report the comparative results in Table 3, and highlight the best results in boldface. The performance improvements of the proposed TME are compared with the best of these baseline methods, marked by the asterisk. We have the following observations from Table 3:

1) We observe that the Random method performs poorly as expected. This is mainly because both datasets have over 100 categories. There is a very high chance of not annotating venues accurately if we randomly choose

Table 3. Performance Comparison in Terms of Macro-F1 and Micro-F1.

Datasets	Methods	mark off x% venues					
		10		20		30	
		Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
TKY	Random	0.91%	1.10%	0.55%	0.70%	0.78%	1.10%
	Majority	0.21%	11.84%	0.23%	12.97%	0.22%	13.00%
	EP [41]	0.68%	11.77%	1.07%	16.43%	1.06%	15.06%
	STES [40]	2.74%	19.06%*	3.28%	19.04%	2.90%	19.56%*
	CARA [28]	1.95%	18.12%	1.83%	17.76%	2.02%	18.49%
	LCE [34]	7.22%	9.57%	8%	12.27%	6.82%	9.05%
	MC-TEM [50]	10.88%	17.33%	11.68%	19.08%*	11.54%	19.03%
	CATAPE [32]	11.78%*	17.33%	12.9%*	18.6%	12.02%*	17.21%
	TME-1	15.75%	23.53%	15.14%	24.18%	13.65%	22.49%
	TME-2	14.54%	23.22%	12.46%	24.73%	11.90%	22.78%
	TME	**16.41%**	**28.63%**	**16.55%**	**29.98%**	**14.38%**	**26.91%**
	Improvements	39.3%	50.21%	28.29%	57.13%	19.63%	37.58%
NYC	Random	0.60%	0.71%	0.69%	0.86%	0.41%	0.60%
	Majority	0.01%	0.97%	0.02%	1.49%	0.02%	1.36%
	EP [41]	0.97%	6.04%	0.87%	7.76%	1.45%	14.04%*
	STES [40]	2.41%	13.13%*	2.17%	12.62%*	2.65%	13.68%
	CARA [28]	1.25%	12.87%	1.08%	11.73%	1.08%	11.64%
	LCE [34]	3.88%	7.08%	4.04%	7.41%	4.52%	7.98%
	MC-TEM [50]	5.94%	8.9%	6.04%	9.25%	6.26%	9.28%
	CATAPE [32]	8.68%*	12.28%	7.72%*	11.51%	8.19%*	12.7%
	TME-1	8.86%	11.37%	7.99%	10.01%	8.47%	10.49%
	TME-2	7.01%	9.09%	7.43%	8.55%	7.35%	8.72%
	TME	**10.37%**	**16.05%**	**10.44%**	**14.98%**	**9.33%**	**14.36%**
	Improvements	19.47%	22.24%	35.23%	18.7%	13.92%	2.28%

a category as the label. The Majority method improves the values of Micro-F1 significantly, as the number of venues of each category is not uniformly distributed and a few categories account for the labels of most venues. Moreover, since it only recommends one category label for venues, this method also obtains the worst values of Macro-F1.

2) EP performs better than Random and Majority, as it extracts several population features to depict venues and makes semantic annotation accordingly. But it does not model the check-in trajectories to capture semantic relationships among venues. STES adopts the embedding methods to model the venue sequences while CARA leverages the RNN framework to model the sequences. They both perform better than EP, indicating that it is effective to take the latent venue representations as the classification features, as these representations could retain venue semantics. But they perform worse than the proposed TME-1, as STES and CARA do not consider the category context in the check-in sequences which reflect aggregated semantics to learn venue representations.

3) MC-TEM, LCE, and CATAPE are trajectory embedding methods, and obtain decent performance, since they can learn representations of venues and categories that preserve certain semantics. Compared with MC-TEM, LCE and CAPATE directly model the categorical information of venues, where LCE minimizes the distances between

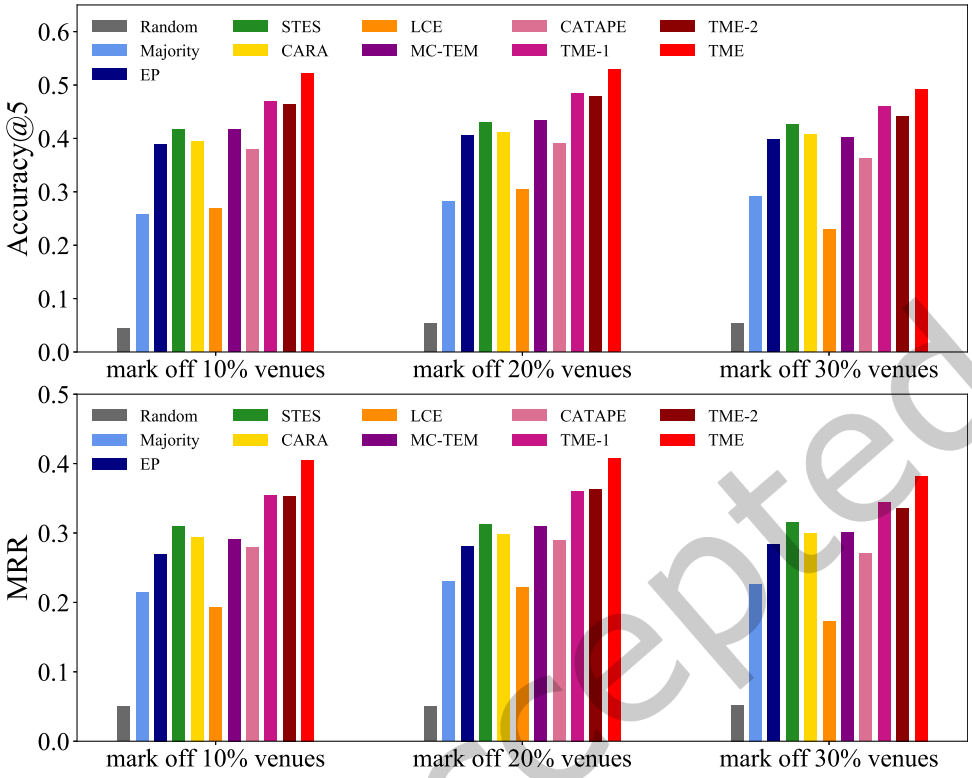


Fig. 3. Performance comparison in terms of Accuracy and MRR.

venues and their corresponding categories in the embedding space and CATAPE predicts the contextual categories given both the target category and the target venue. All these methods perform worse than the proposed TME, as they fail to model the relatedness among categories according to the predefined category hierarchical structure.

4) The proposed TME performs the best. For example, compared with CATAPE, TME achieves an average improvement of 60.9% on the TKY data, and 24.6% on the NYC data, in terms of Micro-F1 for the various percentages of test venues. Further, the paired t-tests are conducted among TME and these baseline methods, and it concludes that the improvement of TME over these baseline methods is of statistical significance with p value < 0.01 [17]. The results demonstrate that TME is more effective than the other competitors for semantic venue annotation, and verify that **RQ1** can be positively answered.

In the real scenario, we may care about not only whether the predicted category is accurate but also where the correct category is placed in the ranking list. We further evaluate the performance of all these methods in terms of Accuracy and MRR, and report the results of the TKY data in Fig. 3. Similar results are observed on the NYC data, and are omitted here. Based on the results, we observe that the proposed TME outperforms the state-of-the-art methods evidently, which further provides a positive answer to **RQ1**.

5.3.2 Model analyses. We design two variants of TME to verify the effectiveness of our proposal. **TME-1** removes the tree-guided multi-task learning component from TME, which is to evaluate the effects of the tree-guided multi-task learning component on improving the performance of semantic annotation. **TME-2** sets λ_1 at 0, which is to evaluate the effects of the category hierarchy on the prediction performance. We record the comparison

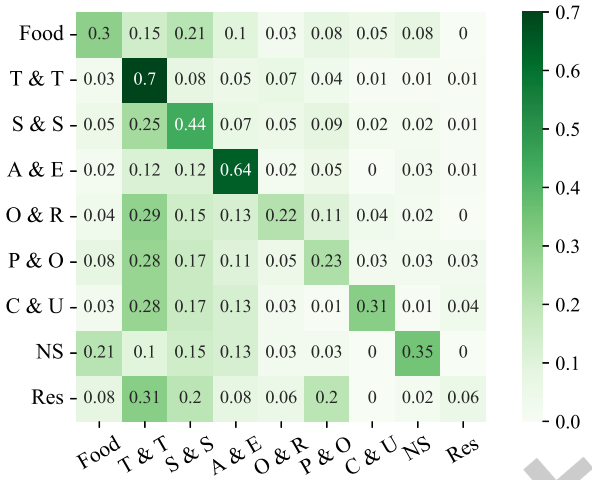


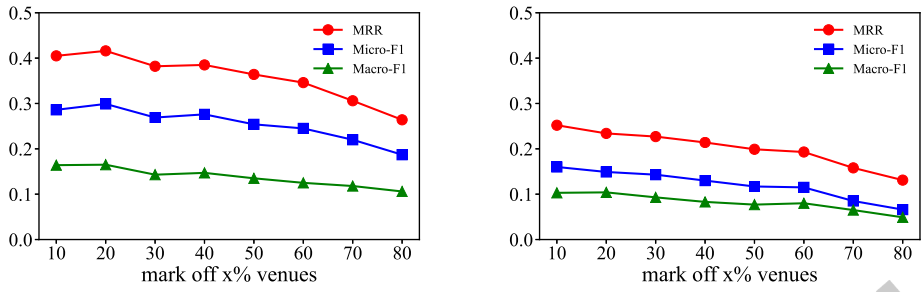
Fig. 4. Comparing real labels (left) and the output from TME (bottom). The categories from left to right (top to bottom) are: Food (*Food*), T & T (*Travel & Transport*), S & S (*Shop & Service*), A & E (*Arts & Entertainment*), O & R (*Outdoors & Recreation*), P & O (*Professional & Other Places*), C & U (*College & University*), NS (*Nightlife Spot*) and Res (*Residence*).

results in Table 3 and Fig. 3, and observe that the performance of TME-1 is better than MC-TEM that merely models the venue context, which provides a positive answer to **RQ2**. Moreover, TME-2 performs worse than TME-1, indicating that it is not enough to simply incorporate the label information into the check-in embedding methods. Further, TME outperforms TME-1, as it leverages the tree-guided multi-task learning to jointly model the label information of venues and the category relatedness. Therefore, a positive answer to **RQ3** can be formed.

Moreover, we go through the performance of TME for the task of semantic venue annotation according to the nine top-layer categories. In specific, we compare the real top-layer categories of venues with the output of TME on the TKY data, and report the results in Fig. 4. Evidently, the majority of venues with category *Travel & Transport* could be labeled accurately, as they have regular and distinguishable feature patterns. Most of the errors are due to *Shop & Service*, *Food*, and *Professional & Other Places* being misclassified as *Travel & Transport* by the proposed TME. The reasons lie in that some venues associated with these categories have similar check-in behaviors to those venues of category *Travel & Transport*.

We further examine the performance of TME by changing the number of marked off venues in the dataset. We mark off the number of all the venues from 10% to 80% via replacing their categories with the "NULL" tag and predict categories for those venues with the "NULL" tag. As shown in Fig. 5, we observe that the performance of TME drops with the increasing number of marked off venues. It is mainly because the less venues with categorical information added into the training data, the harder it is to capture the semantic relations between venues and the corresponding categories.

5.3.3 Case studies. MC-TEM mainly models the venue context from these check-in trajectories, while TME-1 models both the venue context and the category context from trajectories, and TME models both check-in trajectories and the category hierarchy. We respectively list the top-10 categories with the best performance in terms of Micro-F1 for MC-TEM, TME-1 and TME in Fig. 6. We have the following observations from Fig. 6: 1) All these methods achieve stable and satisfactory performance on those categories related to *Travel & Transport*, such as *Airport Gate* and *Light Rail Station*, as venues related to them usually have typical check-in patterns. 2)



(a) TKY (b) NYC

Fig. 5. Performance of TME marking off venues for different numbers.

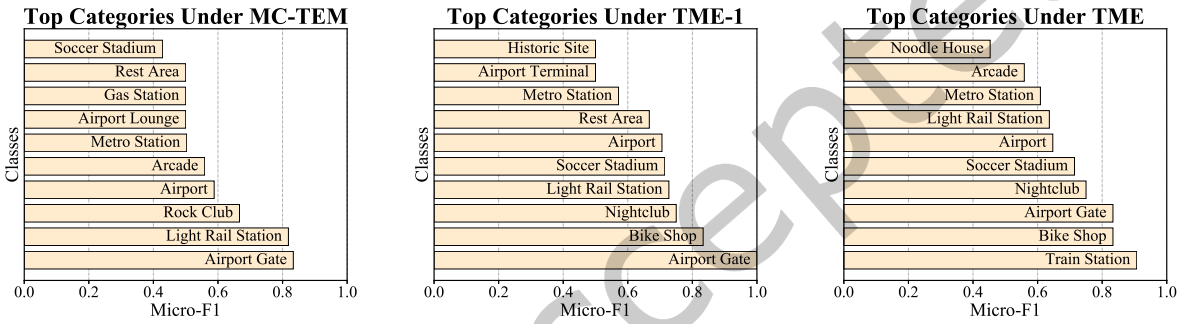


Fig. 6. Top-10 categories with the best performance of MC-TEM, TME-1 and TME.

TME-1 outperforms MC-TEM, as it models the category context additionally. For example, two different venue sequences $l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow l_4$ and $l_5 \rightarrow l_6 \rightarrow l_7 \rightarrow l_8$ may correspond to the same category sequence *Bus Stop* → *Noodle House* → *Bus Station* → *BBQ Joint*. Since venues l_3 and l_7 do not share a similar venue context, MC-TEM tends to learn different embeddings for them. But the two venues have the same category context. TME-1 models both the venue context and the category context, whereby the semantic relations encoded in the category context could enrich and enhance venue embeddings, boosting the performance of venue annotation. 3) We find that the top-10 best performing categories of TME occur frequently in the set of check-in trajectories. For instance, the venues associated with the top-10 categories of TME occur 3,458 times in the TKY dataset, while those associated with the top-10 categories of TME-1 occur 1,200 times. 4) TME enjoys significant improvement on the category of *Train Station*. To explore the reason of improvement, we fetch all test venues with the *Train Station* label and find that many venues have less than 100 check-ins. It is hard for MC-TEM and TME-1 to learn meaningful semantic relations between such sparse venues and their categories, as they only model the check-in trajectories, yielding poor performance. However, TME considers the label information of venues and the category hierarchy, and leverages the semantic relations among categories to guide venue embedding learning. Therefore, TME could encode more semantics into sparse venues and improve the classification performance for such venues accordingly.

5.3.4 Parameter tuning and sensitivity. We have three parameters (context window size, embedding size D and weight λ_1) in the proposed TME. Fig. 7 shows the performance of TME with varying parameters in terms of

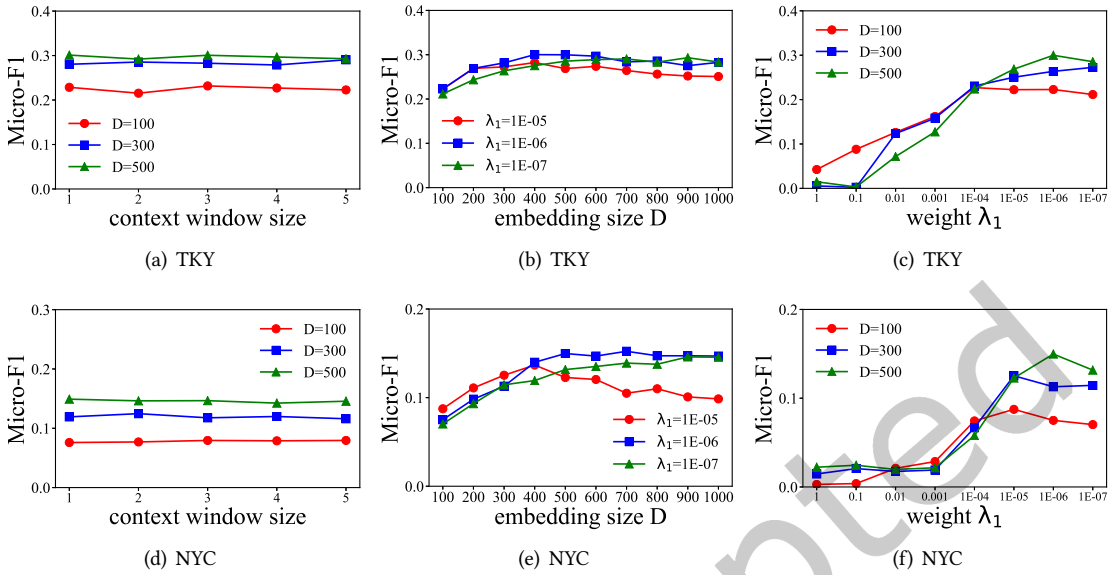


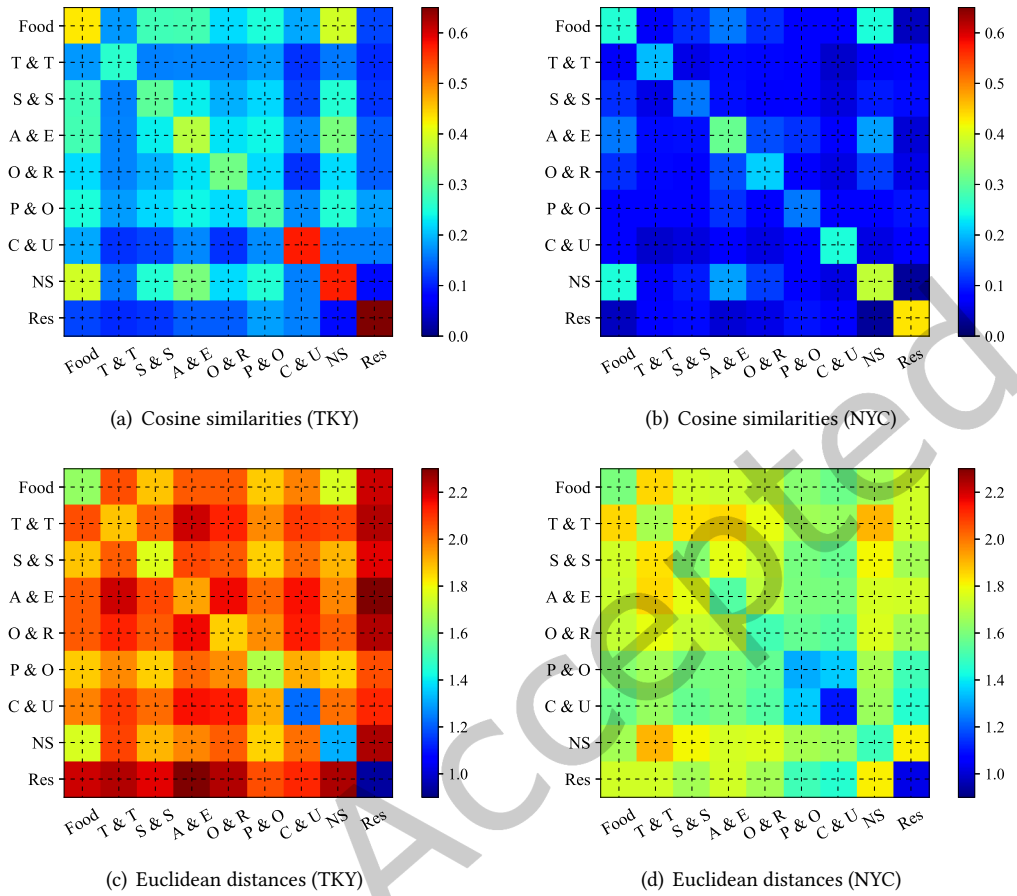
Fig. 7. Effect of varying parameters in TME.

Micro-F1 with $x = 20$. We first fix the embedding size and the weight, and vary the context window size from 1 to 5. We observe from Fig. 7 (a) and (d) that the performance of TME remains relatively stable on both datasets. Next we fix the context window size and the weight, and vary the dimension D of the feature space from 100 to 1,000. The values of Micro-F1 on both datasets are shown in Fig. 7 (b) and (e). Evidently, the values of Micro-F1 improve when we increase the embedding size D from 100 to 500, and then remain relatively stable when increasing it further. Finally, we fix the embedding size and the context window size, and vary λ_1 (which is the weight of the tree-guided constraint) from 1 to 10^{-7}. Observed from Fig. 7 (c) and (f), the values of Micro-F1 have an obvious improvement when λ_1 decreases from 1 to 10^{-5}, and then remain stable on both datasets.

5.4 Qualitative Analyses of Embedding Vectors

Our embedding vectors are designed to retain venue semantics so that venue categories are distinguishable. We set the embedding size at 500 and obtain category representations in both TKY and NYC datasets. To get a qualitative impression of these embeddings, we visualize the relations between each pair of top-layer categories in the form of heatmaps. Specifically, we calculate the cosine similarity for any two categories based on their embedding vectors, and aggregate them to compute the mean values between two top-layer categories. Different from the cosine metric which measures the in-between angle of two vectors, the Euclidean distance focuses on the magnitude of difference between two vectors. Therefore, we also calculate the mean Euclidean distance for each pair of top-layer categories. We report the results in terms of cosine similarity and Euclidean distance on both datasets in Fig. 8, and observe several significant trends.

From Fig. 8 (a) and (b), we observe that the mean cosine similarities of the intra-category embedding vectors are the largest, which means that the representations of categories belonging to the same top-layer category show similar semantics. Further, *Residence* and *College & University* have small inter-category similarities and large intra-category similarities, implying that the children of the two categories are the most compact in the embedding space. The results appear reasonable, as venues related to *Residence* and *College & University* are



(a) Cosine similarities (TKY)

(b) Cosine similarities (NYC)

(c) Euclidean distances (TKY)

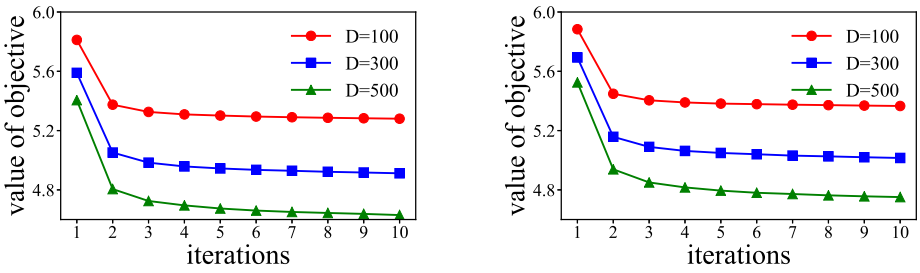
(d) Euclidean distances (NYC)

Fig. 8. Heatmaps of mean cosine similarities and Euclidean distances for category representations, where the categories from left to right (top to bottom) are: Food (*Food*), T & T (*Travel & Transport*), S & S (*Shop & Service*), A & E (*Arts & Entertainment*), O & R (*Outdoors & Recreation*), P & O (*Professional & Other Places*), C & U (*College & University*), NS (*Nightlife Spot*) and Res (*Residence*).

more concentrated in a small district and have evident characteristics. *Outdoors & Recreation*, *Professional & Other Places*, *Travel & Transport*, and *Shop & Service* have moderate intra- and inter-category similarities, which indicates that the children of these categories are widely distributed in the embedding space. Moreover, categories including similar or overlapping venues usually have large inter-category similarities, such as *Food-Nightlife Spot*, and *Arts & Entertainment-Nightlife Spot*. Similar tendencies exist with the mean Euclidean distances between categories in Fig. 8 (c) and (d).

5.5 Efficiency Analyses

We learn the optimal parameters of TME using an iterative method. Therefore, we need to guarantee that the objective of TME could quickly reach a stationary point. Here we vary the number of iterations from 1 to 10 and report the values of the objective on both datasets in Fig. 9. Evidently, when the number of iterations



(a) TKY (b) NYC

Fig. 9. Efficiency performance of TME.

Table 4. Runtime of TME (Unit: Second).

	D=100	D=200	D=300	D=400	D=500
TKY	9.5	13.4	17.6	21.3	25.1
NYC	16.0	21.7	27.1	32.8	39.0

increases, the value of objective starts to decline sharply and becomes steady after about 8 iterations. TME updates these representations of venues and categories including **E**, **L** and **C** at each iteration. Thus the runtime of each iteration is determined by the dimensionality of embedding space D. We show the runtime of an iteration with $D = 100, 200, 300, 400, 500$ in Table 4. Obviously, the runtime with $D = 500$ is the largest, and that with $D = 100$ is the smallest, indicating that there is a positive correlation between the runtime and the embedding size D. Moreover, the runtime on the NYC data is larger than that on the TKY data with the same D, because the number of venues and categories of the NYC data is larger.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we present a Tree-guided Multi-task Embedding model to semantically annotate venues. The proposed TME consists of the sequential and categorical embedding component and the tree-guided multi-task learning component. The first component is capable of learning latent representations of venues and categories from users' check-ins, preserving the sequential and categorical patterns of venues. The second one considers the relations among multiple annotation tasks according to the hierarchical structure of categories, and learns more discriminative representations of venues and categories. Based upon our proposed TME, venues associated with the same category tend to be close to each other in the embedding space. Finally, we evaluate the proposed TME over two real-world check-in datasets from Foursquare, and compare TME with several state-of-the-art baselines. Experimental results show that TME contributes to improving the performance of semantic venue annotation.

In the future, we plan to use multi-modal data (e.g., images, tweets) of venues to learn better venue representations and boost the performance of semantic annotation. Moreover, as TME focuses on learning venue semantics from check-in sequences, it would be interesting to explore how to make semantic venue annotation for those new venues with limited check-ins.

7 ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61906107, the Shandong Excellent Young Scientists Fund (Oversea) under Grant 2022HWYQ-044, the Natural Science Foundation of Shandong Province of China under Grant No. ZR2021QD007, the Young Scholars Program of Shandong University, and the Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources.

REFERENCES

[1] Mohammad Aliannejadi, Dimitrios Rafailidis, and Fabio Crestani. 2018. A Collaborative Ranking Model with Multiple Location-based Similarities for Venue Suggestion. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. 19–26.

[2] Mohammed Alsuhaibani, Takanori Maehara, and Danushka Bollegala. 2019. Joint Learning of Hierarchical Word Embeddings From a Corpus and a Taxonomy. In *Automated Knowledge Base Construction*.

[3] Francis R Bach. 2008. Consistency of the Group Lasso and Multiple Kernel Learning. *Journal of Machine Learning Research* 9, Jun (2008), 1179–1225.

[4] R. Caruana. 1997. Multitask Learning. *Machine Learning* 28, 1 (1997), 41–75.

[5] Chih-Wei Chang, Yao-Chung Fan, Kuo-Chen Wu, and Arbee LP Chen. 2014. On the Semantic Annotation of Daily Places: A Machine-learning Approach. In *Proceedings of the 4th International Workshop on Location and the Web*. 3–8.

[6] Ahmed Cheikhrouhou, Yousri Kessentini, and Slim Kanoun. 2021. Multi-Task Learning for Simultaneous Script Identification and Keyword Spotting in Document Images. *Pattern Recognition* 113, C (2021), 107832.

[7] Hongshen Chen, Jiashu Zhao, and Dawei Yin. 2019. Fine-grained Product Categorization in E-commerce. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2349–2352.

[8] Ling Chen, Mingrui Han, Hongyu Shi, and Xiaoze Liu. 2021. Multi-context Embedding Based Personalized Place Semantics Recognition. *Information Processing and Management* 58, 1 (2021), 102416.

[9] 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.

[10] Meng Chen, Yan Zhao, Yang Liu, Xiaohui Yu, and Kai Zheng. 2020. Modeling Spatial Trajectories with Attribute Representation Learning. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[11] Meng Chen, Lei Zhu, Ronghui Xu, Yang Liu, Xiaohui Yu, and Yilong Yin. 2021. Embedding Hierarchical Structures for Venue Category Representation. *ACM Transactions on Information Systems* 40, 3 (2021), 1–29.

[12] Yingying Duan, Wei Lu, Weiwei Xing, Peng Bao, and Xiang Wei. 2019. PBEM: A Pattern-Based Embedding Model for User Location Category Prediction. In *Proceedings of the 12th International Conference on Mobile Computing and Ubiquitous Network*. 1–6.

[13] Deborah Falcone, Cecilia Mascolo, Carmela Comito, Domenico Talia, and Jon Crowcroft. 2014. What Is This Place? Inferring Place Categories through User Patterns Identification in Geo-tagged Tweets. In *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*. 10–19.

[14] Jianping Fan, Tianyi Zhao, Zhenzhong Kuang, Yu Zheng, Ji Zhang, Jun Yu, and Jinye Peng. 2017. HD-MTL: Hierarchical Deep Multi-task Learning for Large-scale Visual Recognition. *IEEE Transactions on Image Processing* 26, 4 (2017), 1923–1938.

[15] Dehong Gao, Wenjing Yang, Huiling Zhou, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep Hierarchical Classification for Category Prediction in E-commerce System. *arXiv preprint arXiv:2005.06692* (2020).

[16] 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* 8, 1 (2016), 1–24.

[17] David Hull. 1993. Using Statistical Testing in the Evaluation of Retrieval Experiments. In *Proceedings of the 16th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 329–338.

[18] 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*. 163–172.

[19] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in Neural Information Processing Systems* 27 (2014), 2177–2185.

[20] Yanhui Li, Xiangguo Zhao, Zhen Zhang, Ye Yuan, and Guoren Wang. 2020. Annotating Semantic Tags of Locations in Location-based Social Networks. *GeoInformatica* 24, 1 (2020), 133–152.

[21] Hefei Ling, Ziyang Wang, Ping Li, Yuxuan Shi, Jiazhong Chen, and Fuhao Zou. 2019. Improving Person Re-identification by Multi-Task Learning. *Neurocomputing* 347 (2019), 109–118.

[22] Hairong Liu, Xingwei Yang, Longin Jan Latecki, and Shuicheng Yan. 2012. Dense Neighborhoods on Affinity Graph. *International Journal of Computer Vision* 98, 1 (2012), 65–82.

[23] 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*. 5261–5268.

[24] Xiao Lu, Yaonan Wang, Xuanyu Zhou, Zhenjun Zhang, and Zhigang Ling. 2016. Traffic Sign Recognition via Multi-Modal Tree-Structure Embedded Multi-Task Learning. *IEEE Transactions on Intelligent Transportation Systems* 18, 4 (2016), 960–972.

[25] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. 2021. A Survey on Deep Learning for Human Mobility. *Comput. Surveys* 55, 1 (2021), 1–44.

[26] Minnan Luo, Xiaojun Chang, Liqiang Nie, Yi Yang, Alexander G Hauptmann, and Qinghua Zheng. 2017. An adaptive semisupervised feature analysis for video semantic recognition. *IEEE transactions on cybernetics* 48, 2 (2017), 648–660.

[27] Mingqi Lv, Ling Chen, Zhenxing Xu, Yinglong Li, and Gencai Chen. 2016. The Discovery of Personally Semantic Places Based on Trajectory Data Mining. *Neurocomputing* 173 (2016), 1142–1153.

[28] 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 and Development in Information Retrieval*. 555–564.

[29] Kaidi Meng, Haojie Li, Zhihui Wang, Xin Fan, Fuming Sun, and Zhongxuan Luo. 2017. A Deep Multi-modal Fusion Approach for Semantic Place Prediction in Social Media. In *Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes*. 31–37.

[30] 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.

[31] 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. *Information Processing and Management* 57, 2 (2020), 102151.

[32] 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 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 173–176.

[33] Liling Tan, Maggie Yundi Li, and Stanley Kok. 2020. E-Commerce Product Categorization via Machine Translation. *ACM Transactions on Management Information Systems* 11, 3 (2020), 1–14.

[34] 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 2017 International Conference on Neural Information Processing*. 710–720.

[35] Xiaojie Wu, Ling Chen, Mingqi Lv, Mingrui Han, and Gencai Chen. 2017. Cost-sensitive Semi-supervised Personalized Semantic Place Label Recognition Using Multi-context Data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–14.

[36] Caixia Yan, Qinghua Zheng, Xiaojun Chang, Minnan Luo, Chung-Hsing Yeh, and Alexander G Hauptman. 2020. Semantics-preserving graph propagation for zero-shot object detection. *IEEE Transactions on Image Processing* 29 (2020), 8163–8176.

[37] Cheng Yang, Maosong Sun, Wayne Xin Zhao, Zhiyuan Liu, and Edward Y Chang. 2017. A Neural Network Approach to Jointly Modeling Social Networks and Mobile Trajectories. *ACM Transactions on Information Systems* 35, 4 (2017), 1–28.

[38] 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 Web Conference*. 2147–2157.

[39] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory Cultural Mapping Based on Collective Behavior Data in Location-based Social Networks. *ACM Transactions on Intelligent Systems and Technology* 7, 3 (2016), 30.

[40] Jing Yang and Carsten Eickhoff. 2018. Unsupervised Learning of Parsimonious General-purpose Embeddings for User and Location Modeling. *ACM Transactions on Information Systems* 36, 3 (2018), 32.

[41] 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*. 520–528.

[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] Ming Yuan and Yi Lin. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68, 1 (2006), 49–67.

[45] Daniel Yue Zhang, Dong Wang, Hao Zheng, Xin Mu, Qi Li, and Yang Zhang. 2017. Large-scale Point-of-interest Category Prediction Using Natural Language Processing Models. In *2017 IEEE International Conference on Big Data (Big Data)*. 1027–1032.

[46] 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.

[47] 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 Companion*. 153–162.

[48] 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 Transactions on Knowledge Discovery from Data* 12, 2 (2018), 1–21.

[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 Transactions on Knowledge and Data Engineering* 28, 8 (2016), 1945–1958.

[51] Yin Zhu, Erheng Zhong, Zhongqi Lu, and Qiang Yang. 2013. Feature Engineering for Semantic Place Prediction. *Pervasive and Mobile Computing* 9, 6 (2013), 772–783.

APPENDIX

A OPTIMIZATION

A.1 Calculating $\mathbf{L}$ with $\mathbf{E}$ and $\mathbf{C}$ fixed

With $\mathbf{E}$ and $\mathbf{C}$ fixed, we first take the derivative of Γ regarding $\mathbf{L}$ as follows,

$$\frac{\partial \Gamma}{\partial \mathbf{L}} = -2(\mathbf{M}^l - \mathbf{L}\mathbf{E}^{\mathrm{T}})\mathbf{E} - 2(\mathbf{M}^c - \mathbf{L}\mathbf{C}^{\mathrm{T}})\mathbf{C} + 2\lambda\mathbf{L}. \tag{10}$$

By setting Equation (10) to zero, it can be derived that,

$$\mathbf{L} = (\mathbf{M}^l\mathbf{E} + \mathbf{M}^c\mathbf{C})(\mathbf{E}^{\mathrm{T}}\mathbf{E} + \mathbf{C}^{\mathrm{T}}\mathbf{C} + \lambda\mathbf{I})^{-1}, \tag{11}$$

where $\mathbf{I} \in \mathbb{R}^{D\times D}$ is an identity matrix. Because of the inverse operation, we need to ensure that Equation (11) could be solved. Fortunately, in terms of the definition of positive-definite matrix, $(\mathbf{E}^{\mathrm{T}}\mathbf{E} + \mathbf{C}^{\mathrm{T}}\mathbf{C} + \lambda\mathbf{I})$ is positive definite and invertible.

A.2 Calculating $\mathbf{E}$ with $\mathbf{L}$ and $\mathbf{C}$ fixed

Similarly, we fix $\mathbf{L}$ and $\mathbf{C}$, and calculate the derivative of Γ regarding $\mathbf{E}$ as follows,

$$\frac{\partial \Gamma}{\partial \mathbf{E}} = -2({\mathbf{M}^l}^{\mathrm{T}} - \mathbf{E}\mathbf{L}^{\mathrm{T}})\mathbf{L} + 2(\mathbf{E}\mathbf{C}^{\mathrm{T}} - \mathbf{Y})\mathbf{C} + 2\lambda\mathbf{E}. \tag{12}$$

By setting Equation (12) to zero, we have

$$\mathbf{E} = ({\mathbf{M}^l}^{\mathrm{T}}\mathbf{L} + \mathbf{Y}\mathbf{C})(\mathbf{L}^{\mathrm{T}}\mathbf{L} + \mathbf{C}^{\mathrm{T}}\mathbf{C} + \lambda\mathbf{I})^{-1}, \tag{13}$$

where $(\mathbf{L}^{\mathrm{T}}\mathbf{L} + \mathbf{C}^{\mathrm{T}}\mathbf{C} + \lambda\mathbf{I})$ is also positive definite and invertible.

A.3 Calculating $\mathbf{C}$ with $\mathbf{E}$ and $\mathbf{L}$ fixed

Note that $\sum_{c\in C} e_c||\mathbf{C}_{\mathcal{G}_c}||_{2,1}$ in Equation (8) is differentiable. We replace it via an equivalent formulation following [3],

$$\lambda_1\Big(\sum_{c\in C} e_c||\mathbf{C}_{\mathcal{G}_c}||\Big)^2. \tag{14}$$

But Equation (14) still cannot be solved easily. Therefore, we need another variational formulation to facilitate optimization. Based on the Cauchy-Schwarz inequality, we derive the following inequality,

$$\Big(\sum_{c\in C} e_c||\mathbf{C}_{\mathcal{G}_c}||\Big)^2 \le \sum_{c\in C}\sum_{d=1}^{D} \frac{e_c^2||\mathbf{C}_{\mathcal{G}_c}^d||_2^2}{\mathbf{q}_{c,d}}, \tag{15}$$

where $\mathbf{C}_{\mathcal{G}_c}^d$ denotes the dth column vector of the group matrix $\mathbf{C}_{\mathcal{G}_c}$, and $\sum_c\sum_d \mathbf{q}_{c,d} = 1, \mathbf{q}_{c,d} \ge 0, \forall c, d$. It is worthy noting that the equality holds when

$$\mathbf{q}_{c,d} = \frac{e_c||\mathbf{C}_{\mathcal{G}_c}^d||_2^2}{\sum_{c\in C}\sum_{d=1}^{D} e_c||\mathbf{C}_{\mathcal{G}_c}^d||_2^2}. \tag{16}$$

Algorithm 1: optimization of TME model

Require: $\mathbf{M}^l$, $\mathbf{M}^c$,$\mathbf{Y}$, D, λ_1, λ
Ensure: $\mathbf{E}$, $\mathbf{L}$, $\mathbf{C}$
1: initialize $\mathbf{E}$, $\mathbf{L}$, and $\mathbf{C}$
2: **while** not converge **do**
3: fixing $\mathbf{E}$ and $\mathbf{C}$, update $\mathbf{L}$ according to Equation (11),
4: fixing $\mathbf{L}$ and $\mathbf{C}$, update $\mathbf{E}$ according to Equation (13),
5: fixing $\mathbf{C}$, update $\mathbf{q}_{c,d}$ according to Equation (16),
6: updating $\mathbf{Q}$ according to Equation (17),
7: fixing $\mathbf{L}$, $\mathbf{E}$ and $\mathbf{Q}$, update $\mathbf{C}$ according to Equation (19).
8: **end while**

To facilitate the computation of the derivative of Γ with respect to $\mathbf{C}_i$ for category c_i, we define a new matrix $\mathbf{Q} \in \mathbb{R}^{D\times D}$,

$$\sum_{c\in C}\sum_{d=1}^{D}\frac{e_c^2||\mathbf{C}^d_{\mathcal{G}_c}||_2^2}{\mathbf{q}_{c,d}} = \sum_{c\in C}\sum_{d=1}^{D}\sum_{c_i\in\mathcal{G}_c}\frac{e_c^2}{\mathbf{q}_{c,d}}||\mathbf{C}^d_{c_i}||_2^2,$$
$$\mathbf{Q}^i_{dd} = \sum_{c\in C, c_i\in\mathcal{G}_c}\frac{e_c^2}{\mathbf{q}_{c,d}}, \tag{17}$$
$$\mathbf{Q} = \sum_{i}^{N_c}\mathbf{Q}^i.$$

Combining the above equations, we derive that optimizing Γ regarding $\mathbf{C}$ is the same as optimizing the following convex objective,

$$\begin{aligned}\min_{\mathbf{C},\mathbf{Q}}\Gamma =& ||\mathbf{M}^l - \mathbf{L}\mathbf{E}^{\mathrm{T}}||_F^2 + ||\mathbf{M}^c - \mathbf{L}\mathbf{C}^{\mathrm{T}}||_F^2 \\ &+ ||\mathbf{Y} - \mathbf{E}\mathbf{C}^{\mathrm{T}}||_F^2 + \lambda_1\mathbf{C}\mathbf{Q}\mathbf{C}^{\mathrm{T}} \\ &+ \lambda(||\mathbf{L}||_F^2 + ||\mathbf{E}||_F^2 + ||\mathbf{C}||_F^2).\end{aligned} \tag{18}$$

We set the derivative of Γ with respect to $\mathbf{C}$ to zero and obtain

$$\mathbf{C} = (\mathbf{M}^{c\mathrm{T}}\mathbf{L} + \mathbf{Y}^{\mathrm{T}}\mathbf{E})(\mathbf{L}^{\mathrm{T}}\mathbf{L} + \mathbf{E}^{\mathrm{T}}\mathbf{E} + \lambda_1\mathbf{Q} + \lambda\mathbf{I})^{-1}. \tag{19}$$

Algorithm 1 shows the pseudo code of training the TME model. Based on users' check-in trajectories, we first construct the venue co-occurrence positive PMI matrix $\mathbf{M}^l$ using Equation (1), the venue category positive PMI matrix $\mathbf{M}^c$ using Equation (3), and the venue category label matrix $\mathbf{Y}$. Then we use the venue category hierarchy to build $\mathcal{G}_c$ for each category c. Taking $\mathbf{M}^l$, $\mathbf{M}^c$ and $\mathbf{Y}$ as inputs, we could learn the representations of venues and categories.