Human Mobility Identification by Deep Behavior Relevant Location Representation

Tao Sun[1,2], Fei Wang[1], Zhao Zhang[1], Lin Wu[1], and Yongjun Xu[1](✉)

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{suntao,wangfei,zhangzhao2021,wulinshuxue,xyj}@ict.ac.cn

[2] School of Computing Science and Technology,
University of Chinese Academy of Sciences, Beijing, China

Abstract. This paper focuses on Trajectory User Link (TUL), which aims at identifying user identities through exploiting their mobility patterns. Existing TUL approaches are based on location representation, a way to learn location associations by embedding vectors that can indicate the level of semantic similarity between the locations. However, existing methods for location representation don't consider the semantic diversity of locations, which will lead to a misunderstanding of the semantic information of trajectory when linking anonymous trajectories to candidate users. To solve this problem, in this paper, we propose Deep Behavior Relevant Location representation (DBRLr) to map the polysemous locations into distinct vectors, from the perspective of users' behavior to reflect the semantic polysemy of locations. To learn this representation, we build a Location Prediction-based Movement Model (LP-based MM), which learns user behavior representation at each visited location from a large history trajectory corpora. LP-based MM considers both Continuity and Cyclicity characteristics of user's movement. We employ the combination of the intermediate layer representation in LP-based MM as DBRLr. An effective recurrent neural network is used to link anonymous trajectories with candidate users. Experiments are conducted on two real-world datasets, and the result shows that our method performs beyond existing methods.

Keywords: Human mobility identification · Trajectory-user link · Location representation · Polysemous location

1 Introduction

The plentiful location-based applications make it possible to accumulate lots of users' movement data. Massive anonymous trajectories, which we do not know who created them, are collected, bringing in many problems to trajectory-based analysis. Trajectory User Link (TUL) [8] aims to solve this problem, to identify anonymous trajectories and associate them with candidate users. Because the

A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13246, pp. 439–454, 2022.
https://doi.org/10.1007/978-3-031-00126-0_33

user's behavior information is difficult to fully analyze, TUL is still a challenging problem. In recent years, there have been many works focused on TUL [7,8,13, 20,22,28,29].

Existing TUL approaches are based on location representation. Similar to word embedding [14] in natural language processing and node embedding in graph learning [9], the location representation learns embedding vectors that can indicate semantic similarity between the locations from large historical trajectory corpora. By location representation, the original anonymous trajectory, represented by longitude and latitude, can be converted into a sequence composed of semantic locations, which benefits a better understanding of the whole trajectory information. Existing location representation methods consider sequential information [1,11] and spatial information [5,31]. All those approaches are based on a single semantic location representation, establishing a one-to-one correspondence between locations and their semantic vector representations.

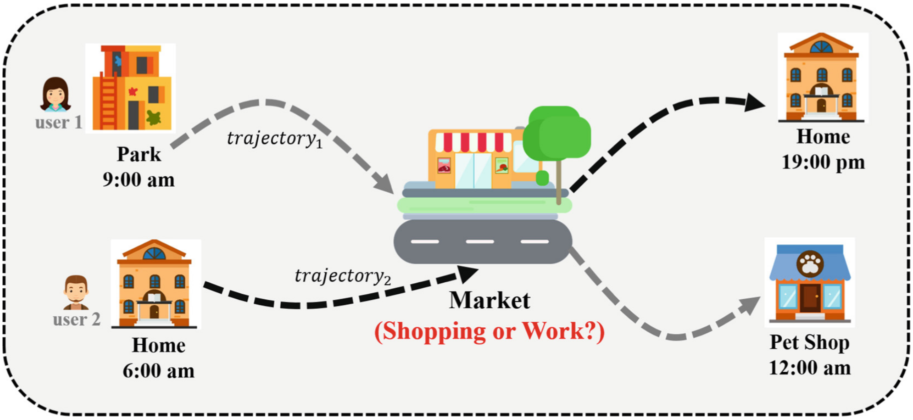


Fig. 1. A demonstration of polysemous location.

Nevertheless, we hold that the semantic information of locations is polysemous. Precisely, the same location unfolds varying effects for different users at different time. To illustrate our motivation more precisely, we set up a simple example. As shown in Fig. 1, there are two trajectories. Suppose the two trajectories belong to two users (In fact, we do not know this message because those trajectories are anonymous). Both user 1 and user 2 have reached a market. We can conjecture from their complete trajectories that they have completely different behaviors on the market. In this example, the exact location (market) generates specific semantic information in different trajectories, which is a universal phenomenon. However, existing TUL methods ignore the polysemy of locations, which would lead to a biased understanding of trajectory patterns and misjudge the candidate user corresponding to this trajectory.

To solve this problem, we propose **DBRLr**, **D**eep **B**ehavior **R**elevant **L**ocation representation, which learns the semantic information of user's behavior on location to represent the polysemous location embedding. To learn this representation, we build a **L**ocation **P**rediction-based **M**ovement **M**odel (LP-based MM) to learn user behavior from a large number of historical trajectories. LP-based MM considers **2C** characteristics of users' behavior: continuity and cyclicity, and employs the combination of the intermediate layer representation as DBRLr. After that, we establish the connection between anonymous trajectory and candidate users based on a deep recurrent neural network called Linker. The Linker takes the representation of the anonymous trajectory as input and outputs the probability of each candidate user. We conduct experiments on three real-world datasets, and the result shows that our method performs beyond existing approaches. Our contributions are as follow:

- We propose **DBRLr**, a polysemous location representation method. DBRLr utilizes the behavior characteristics of the user's visiting location to represent the location. As a result, the same location can reflect different semantic information.
- We establish LP-based **MM** to depict the user movement pattern on trajectories. LP-based MM learns from numerous historical trajectory corpora and embeds locations dynamically based on complete trajectory information.
- We employ DBRLr for **TUL** problem and conduct experiments on three real-world datasets. Extra trajectories of history is used for training MM. The results show that employing MM for location representation improves TUL performance. Our source codes are publicly available at https://github.com/taos123/TUL_by_DBRLr.

2 Related Works

2.1 Trajectory Classification

TUL is one kind of trajectory classification problem if we regard one user as one category of trajectory classification. Trajectory classification has been widely studied and applied [23]. We mainly introduce two kinds of trajectory classification methods. One is based on trajectory similarity measure metrics such as *Euclidean Distance*, *Hausdorff Distance*, *Dynamic Time Warping Distance (DTW)*, *Longest Common Subsequence (LCSS) Distance*, and *Fréchet Distance*, or trajectory feature extraction method of trajectory [6]. Another is based on deep learning, which employs deep neural networks to learn a trajectory representation [3,8,10,12]. These methods mostly use recurrent neural networks to extract trajectory characteristics and classify trajectories, which can effectively deal with the problem of the uncertain length of the trajectory. In recent years, there are some pertinent trajectory classification approaches for TUL problems, such as TULER [8], TULVAE [29], TULAR [22] and AdattTUL [7]. Reference [8] is the first work to put forward the TUL problem formally and employs location embedding, in which locations represented by longitude and latitude are learned into semantic information representation. Variational autoencoder is employed into TUL in [29] to improve TUL performance. Adversarial neural networks are employed [7] for generating more training trajectories. Both [13] and [22] consider the influence of different locations by attention mechanism.

2.2 Location Representation

A graph-based embedding model is proposed in [24], jointly capturing the sequential effect, geographical influence, temporal cyclic effect, and semantic effect in a unified way. Reference [27] proposes a model to capture the semantics information of place types. The model is based on Word2Vec, which augments the spatial contexts of POI types by using distance and information-theoretic approaches to generate embedding. Reference [5] proposes POI2Vec, a latent representation model, which uses the geographical influence of POIs to learn latent representations. Reference [16] introduces a model called Deepcity, which is based on deep learning. It is used to learn features for user and location profiling. Reference [4] employs SkipGram [15] algorithm to predict a location's context given the location itself, which contains the representation of the location. Reference [30] learns latent representations of places by directly model movements between places with large-scale movement data. Reference [26] proposes an unsupervised machine language translation method to translate location representations across different cities. Reference [21] proposes fine-grained location embedding by leveraging hierarchical spatial information according to the local density of observed data points to overcome the data sparsity problem. Existing work considers location embedding from different perspectives of structural information but does not consider the diversity of location semantics. To the best of our knowledge, it is the first time to propose polysemous location representation in trajectory data mining domains.

3 Methodology

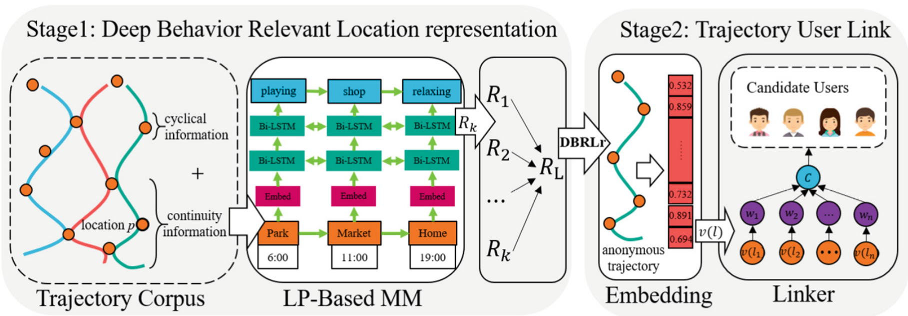


Fig. 2. An overview structure of trajectory user link with deep behavior relevant location representation.

We introduce the technical details of DBRLr for TUL in this section. Our method consists of two main stages: 1) Deep behavior relevant location representation. In this part, we build a LP-based MM based on continuity and cyclicity of user' movement and train it, taking advantage of a lot of history trajectory corpora. We extract the intermediate layer representation of MM as DBRLr. 2) Trajectory-User Link. In this stage, the anonymous trajectory is embedded into semantic vectors by DBRLr. A two-layer biLSTM and a simple classifier are used to capture characteristics and link the anonymous trajectories to candidate users. Figure 2 presents an overview of our method.

3.1 Preliminary

We will present the mathematical notations and problem statement first. Let $U = \{u_1, u_2, \cdots, u_M\}$ be user set. Let $L = \{l_1, l_2, \cdots, l_K\}$ be all of locations that all users visited. For each user, the user's movement through space produces a sequence of locations, which is denoted by $T = \{l_1, l_2, \cdots, l_i, \cdots, l_N\}$, where $l_i \in L$ and N is the length of trajectory. Non-anonymous trajectories will contain user's identity information. For an anonymous trajectory, we need to infer the corresponding user identify.

Given an anonymous trajectory T, the trajectory-user link aims to indicate the most likely user from U that is most likely to produce the anonymous trajectory. TUL learns a mapping function that links trajectories to users: $T \rightarrow U$ [29].

3.2 Empirical Analysis

We first intuitively investigate two problems: Is users' behavior learnable? How to learn a user's behavior in a given location? Due to the restriction of trajectory acquisition technology and privacy protection, users' behaviors in locations are not captured in almost all public trajectory datasets. Though, the user's behaviors are potentially indicated in the user's trajectories. As it is shown in Fig. 3, we summarize users' movement adhering to **2C** principles: **C**ontinuity and **C**yclicity. Continuity means that where a user stays is affected by his previous locations and where he plans to go [18,31]. Cyclicity means that users will be influenced by regular habits when they decide where to go [19,24]. Through the above two characteristics, we can obtain the similarity of users' behavior in locations. Therefore, if we can establish a model to comprehensively consider the two characteristics based on an extensive trajectory data corpora, we can learn the similar relationship between users' behaviors and get the behavior representation of users on a location.

3.3 Location Prediction Based Movement Model

Inspired by the language model [17], we propose a location prediction-based movement model to learn the users' movement. We first describe the parts of the

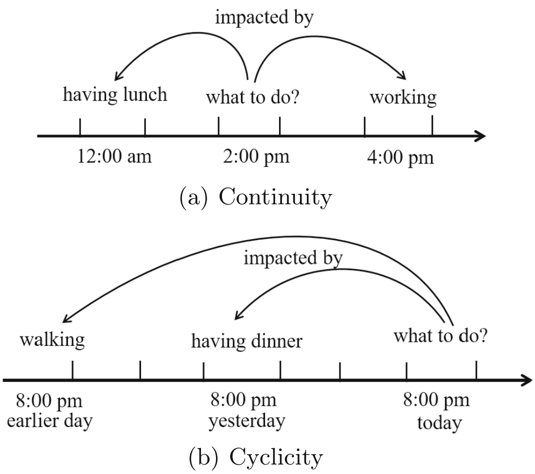


Fig. 3. 2C users' movement principles.

continuity. We can intuitively discover that a user's current location is affected by his historical location. For a trajectory T in historical trajectory corpora, T contains N locations, $(l_1, l_2, \cdots, l_N)$. A forward movement model computes the probability of the trajectory by modeling the probability of location l_k given the preceding locations (l_1, l_2, l_{k-1}). The probability is shown as Eq. 1.

$$p(l_1, l_2, \cdots, l_N) = \prod_{k=1}^{N} p(l_k | l_1, l_2, \cdots, l_{k-1}) \tag{1}$$

At the same time, we also consider the users' visiting backward model, which means the user's current location is affected by the location he will visit later. A backward movement model is similar to a forward movement model, except it runs over the sequence in reverse, predicting the current locations given the future locations. The backward model is shown in Eq. 2.

$$p(l_1, l_2, \cdots, l_N) = \prod_{k=1}^{N} p(l_k | l_{k+1}, l_{k+2}, \cdots, l_N) \tag{2}$$

We employ the location embedding by reference [8] then pass it through L layers of forward LSTMs. At each location l_k, each LSTM layer outputs a context-dependent representation $\overrightarrow{h}_{k,j}^{MM}$ where j denotes the LSTM layer and $j = 1, 2, \cdots, L$. Let $\overrightarrow{h}_{k,L}^{MM}$ denote the top layer LSTM output, which is used to predict p_{k+1} with a softmax layer which is shown in Eq. 3.

$$p(l_k) = \frac{\overrightarrow{h}_{k,L}^{MM}}{\sum_{i=1}^{N} \overrightarrow{h}_{i,L}^{MM}} \tag{3}$$

It can be implemented in an simple way to a forward model, with each backward LSTM layer j in a L layers deep model producing representation $\overleftarrow{h}_{k,j}^{MM}$ of l_k given $(l_{k+1}, l_{k+2}, \cdots, l_N)$.

We employ a biLSTM combining both forward and backward models as continuity parts. We maximize the log likelihood of the forward and backward model by Eq. 4.

$$\begin{aligned}\sum_{k=1}^{N}(\log p(l_k, \cdots, l_{k+1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) \\ + \log p(l_{k+1}, \cdots, l_n; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s))\end{aligned} \tag{4}$$

We consider the cyclicity of movement from global perspective. For all trajectory $\{T_1^k, T_2^k, \ldots, T_n^k\}$ which contain location l_k, we extract locations $\{l_n^{T_j^k}\}$ in the same period with l_k, where $l_j^{T_i^k} \in T_i^k$ and $l_j^{T_i^k}$ is visited in same periods with l_k. Similarly, we need to establish the optimal location probability under this set of periodic correlation sets. The probability is shown as Eq. 5.

$$p(l_1, l_2, \cdots, l_n) = \prod_{k=1}^{N}\prod_{j=1}^{n_k} p(l_k | l_j^{T_n^k}) \tag{5}$$

The cyclicity can also implemented by the same biLSTM. The maximizes the log likelihood of the cyclicity of movement model is shown as formulation 6:

$$\sum_{k=1}^{N}(\log p(l_k; \Theta_x, \widetilde{\Theta}_{LSTM}, \Theta_s) \tag{6}$$

The LP-based MM, considering both continuity and cyclicity, can be trained under jointly maximized log likelihood with formulation 4 and formulation 6.

3.4 Deep Behavior Relevant Location Representations

After training LP-based MM, we use the combination of the intermediate layer representation in biLSTM. For each location l_k, a L-layer LP-based MM computes a set of 2L+1 representation.

$$\begin{aligned}R_k &= \left\{X_k^{MM}, \overrightarrow{h}_{k,j}^{MM}, \overleftarrow{h}_{k,j}^{MM}, \widetilde{h}_{k,j}^{MM} | j = 1, \ldots, L\right\} \\ &= \left\{h_{k,j}^{MM} | j = 0, \ldots, L\right\}\end{aligned} \tag{7}$$

Where $h_{k,0}^{MM}$ is the first location layer and $h_{k,j}^{MM} = \left[\overrightarrow{h}_{k,j}^{MM}; \overleftarrow{h}_{k,j}^{MM}; \widetilde{h}_{k,j}^{MM}\right]$, for each biLSTM layer.

DBRLr collapses all layers in R into a single vector, $DBRLr_k = E(R_k; \Theta_\varepsilon)$. In the simplest case, DBRLr just selects the top layer.

Given a trajectory $T = (l_1, l_2, \cdots, l_N)$, the location l_k in T represented by DBRLr is $v(l_k^T)$.

3.5 Trajectory-User Linker

For the anonymous trajectory datasets to be identified, we first input each anonymous trajectory into the behavior representation layer to obtain the trajectory representation with DBRLr, which are represented as $\left(v(l_1^T), v(l_2^T), \cdots, v(l_N^T)\right)$.

To process the long-term variable-length location sequence, we employ a biLSTM to control input and output of location embedding. For the input trajectory $Tra_j = \{l_1, l_2, \cdots, l_N\}$, let $\{h_1, h_2, \cdots, h_N\}$ denote the output status of biLSTM as $h_t = biLSTM(v(l_t))$. In this way, we can get the every time of the biLSTM outputs. We use a weighted average formula to fuse every time information by Eq. 8.

$$v\left(Tra_j\right) = \sum_{t=1}^{N} a_t h_t \tag{8}$$

where a_t is the weight of location on t, reflecting the influence of this l_t to the whole trajectory. We calculate a_t by the following Eq. 9.

$$a_t = \tanh\left(W_1 h_t + W_2 \overline{h}_s\right) \tag{9}$$

where W_1 and W_2 are parameters to learn and $\overline{h}_s$ is the mean value of $\{h_t | t = 1, 2, \cdots, N\}$. Then we employ fully connection layer with parameters $W_3 \in \mathbb{R}^{N \times M}$ and $b \in \mathbb{R}^{1 \times M}$ to mapping trajectory information to the dimensions of the candidate user set.

$$v\left(Tra_j\right)^{user} = v\left(Tra_j\right) * W_3 + b \tag{10}$$

Let $p\left(u_i | Tra_j\right)$ denote the probability that trajectory Tra_j belongs to user u_i, which is calculated as Eq. 11. The softmax function converts logits into probabilities.

$$p\left(u_i | Tra_j\right) = \frac{v\left(Tra_j\right)^{user}}{\sum_{k=1}^{M} v\left(Tra_j\right)^{user}} \tag{11}$$

To measure the distance from the truth values, we compute softmax cross entropy between logits and labels as Eq. 12. In the training process, our objective is to minimize the loss function.

$$\mathcal{L} = \frac{1}{N} \sum_{j=1}^{N} \left(v\left(u_i\right) - \sum_{i=1}^{M} \log\left(p\left(u_i | Tra_j\right)\right) \right) \tag{12}$$

Finally, given an anonymous trajectory Tra, the corresponding predicted user u_i is calculated.

$$\underset{1<i<M}{\arg\max}\, p\left(u_i | Tra_j\right) = \left\{u_i \in U : p\left(u_i | Tra_j\right)\right\} \tag{13}$$

4 Experiments

In this section, we conduct experiments to evaluate the accuracy of the proposed DBRLr by answering the following three key research questions.

- **Q1**: Does DBRLr outperform the existing TUL baselines in real-world datasets?
- **Q2**: Does the LP-based MM improve the performance of DBRLr? How does DBRLr perform with single-direction movement models?
- **Q3**: Does DBRLr distinguish between polysemous locations? How does the polysemous location representation improve the performance of TUL?

4.1 Datasets

We conduct our experiments on three benchmark datasets [2]: **Gowalla**[1], **Brightkite**[2] and **Foursquare**[3], which are publicly available. Both of them are collected from location-based social networking websites where users share their locations by checking in. The data recorded information such as [**user id, check-in time, longitude, latitude, location id**]. We randomly select a set of users in Gowalla, Brightkite and Foursquare, which are the same number as [29]. The statistics of datasets are summarized in Table 1.

Table 1. Datasets description and statistics

Datasets	$\|U\|$	$\|T\|$	$\|C\|$	$\|Ave\|$
Gowalla	201	19968	1958	99.34
Brightkite	92	19904	471	216.34
Foursquare	300	13281	162	44.27

$|U|$ is the number of users in the datasets. $|T|$ denotes the number of trajectories sets. As we can see $|T| \gg |U|$. $|C|$ is the number of check-in locations. $|Ave|$ represents the average number of check-in locations per trajectory, which is calculated by dividing $|T|$ by $|U|$.

4.2 Baseline Algorithms

We compare our method with both classical trajectory classification methods and TUL approaches. The following baseline models are evaluated.

- **TULER** [8]. TULER uses the check-in location embedding method to reinforce the check-in location information. TULER employs LSTM, GRU, and their variants as the RNN model, which is called: TULER-L, TULER-G, and Bi-TULER. We employ open-source TULER in github[4].

[1] Gowalla: http://snap.stanford.edu/data/loc-Gowalla.html.
[2] Brightkite: http://snap.stanford.edu/data/loc-Brightkite.html.
[3] https://sites.google.com/site/yangdingqi/home.
[4] TULER: https://github.com/gcooq/TUL.

– **TULVAE** [29]. TULVAE learns the human mobility in a neural generative architecture with stochastic latent variables than span hidden states in RNN. We employ open-source TULVAE in github[5].
– **AdattTUL** [7]. AdattTUL is a semi-supervised method, which makes adversarial mobility learning for human trajectory classification, which is an end-to-end framework modeling human moving patterns.
– **TULAR** [13,22]. TULAR considers the influence of different locations and introduces trajectory attention mechanism. TULAR is the state-of-the-art method for TUL. We employ open-source TULAR in github[6].

4.3 Evaluation Metrics

We employ Acc1, Acc5, and macro-F1 as the evaluation metrics, which are the standard metrics of TUL problem [29]. The definitions of those metrics are shown in the following equations.

$$AccK = \frac{\#correctly\ linked\ trajectories\ @K}{\#trajectories} \tag{14}$$

where the $\#correctly\ linked\ trajectories$ @K is the correct users at top K candidates, and $\#trajectories$ is the total number of anonymous trajectories. In addition, because TUL is a multi-classification task, we also need to consider macro-R, macro-P, and macro-f1. The macro-R is the mean of recall value of every classification, and macro-P is the mean of the precision value of every classification. The macro-f1 is defined as follows.

$$macro - F1 = 2 \times \frac{macro - P \times macro - R}{macro - P + macro - R} \tag{15}$$

4.4 Parameter Setup

The training process of our model includes two stages: DBRLr training and Linker training. In the DBRLr training process, we employ a 2-lay bi-LSTM for realizing LP-based MM with input dimension 256. We slice the original trajectory data at 6-hour intervals. We set at least ten training epochs for movement model training. We introduce the regularization method dropout with a dropout rate of 0.1. In the Linker training process, we set the input dimension of Linker to 256. We set the initial learning rate as 0.001, and after 20 to 30 iterations, we reduce the learning rate by half. We use Adam as the optimizer. What's more, we shuffle the training trajectory data set before the initialization of the model. We'll expose the code later in GitHub for reproduction.

[5] TULVAE: https://github.com/AI-World/IJCAI-TULVAE.
[6] TULAR: https://github.com/taos123/TULAR.

4.5 Overall Performance (Q1)

Table 2 exhibits the overall results compared our method with baselines. It can be seen that our method has a significant improvement for TUL. More specifically, on Gowalla and Foursquare, our method is higher than baselines at one percentage point, with over 2% performance improvement in Acc@1. On Brightkite, our method is higher than baselines at nine percentage points, with over 18% performance improvement in Acc@1. This means that we have achieved a recognition accuracy of more than 50%, which will significantly improve TUL availability in practical application scenarios. It can also be seen that our method improves the effect more on Brightkite than on Gowalla and Foursquare.

In has been verified in [29] that the classical trajectory classification method has poor performance for TUL because those classical trajectory classification methods measure the geospatial similarity between trajectories, while TUL needs to find the behavioral similarity between trajectories. There is a slight discrepancy between the results of TULER, TULVAE, AdattTUL, and TULAR. This is because these approaches focus on Linker model improvements to capture trajectory patterns. However, as their input layer, single semantic location representation will cause errors in the semantic information of some locations, resulting in biased recognitions for some trajectories. The improvement of our method benefits from a more accurate understanding of the semantic information of trajectory. DBRLr infers the specific semantic information of each location according to the knowledge learned from the historical trajectory corpora and the context information of the whole trajectory.

4.6 Ablation Experiments (Q2)

In order to explore how LP-based MM impacts the performance of DBRLr, we design five contrast movement models to retrain DBRLr and compare those performances on TUL. The model variants and descriptions are shown in Table 3. We train these above four MMs in the same history trajectory corpora and the same experimental environment. Then we extract location representations from the trained movement model. Applying the above different location representation in TUL with Linker in Sect. 3.4, the performance comparison is shown in Fig. 4.

Figure 4 shows the comparison between different contrast models. We can see that DBRLr with a 2-layer bi-direction movement model achieves the best performance. The location embedding effect without the MM model is generally lower than that with the MM model. Compared with the single-direction movement model, and the bi-direction model can better capture global information of trajectory, which significantly improves the accuracy of TUL. It can also be observed that DBRLr with two-layer performance is better than one layer, indicating that high-level exploring might obtain more practical information from human movement. Nevertheless, this gap is not very obvious. We think the higher layer can capture the information for trajectory. Furthermore, a two-direction of movement model can describe movement patterns with advantages.

Table 2. Performance comparison on Gowalla and Brightkite

Method	Metric								
	Acc@1	Acc@5	macro-F1	Acc@1	Acc@5	Macro-F1	Acc@1	Acc@5	Macro-F1
	Gowalla			Brightkite			Foursquare		
TULER-L	0.4179	0.5789	0.3243	0.4124	0.5688	0.3007	0.5122	0.5911	0.4566
TULER-G	0.4261	0.5795	0.3391	0.4085	0.5731	0.2864	0.5091	0.5887	0.4560
Bi-TULER	0.4267	0.5954	0.3215	0.4195	0.5758	0.3190	0.5388	0.6141	0.4873
TULVAE	0.4435	0.6446	0.3621	0.4540	0.6239	0.3541	0.5428	0.6169	49.22
AdattTUL-G	0.4692	0.6364	0.3726	0.4838	0.6496	0.4269	0.5783	0.6463	0.5364
AdattTUL-L	0.4761	0.6464	0.3774	0.4891	0.6544	0.4335	0.5812	0.6470	0.5385
TULAR-L	0.3265	0.4613	0.2702	0.3012	0.3913	0.2302	0.5881	0.6474	0.5293
TULAR-G	0.3786	0.4928	0.3408	0.4050	0.5338	0.3998	0.5853	0.6521	0.5301
TULAR-B	0.4125	0.5550	0.3432	0.4207	0.6146	0.3659	0.5807	0.6533	0.5358
Our Method	**0.4875**	**0.7227**	**0.4055**	**0.5798**	**0.7750**	**0.5511**	**0.6072**	**0.6933**	**0.5541**

On Gowalla, our method higher than existing methods than 2.39%, 11.80%, 7.45% in terms of Acc@1, Acc@5, F1. On Brightkite, our method higher than existing methods than 18.54%, 18.42%, 27.12% in terms of Acc@1, Acc@5, F1. On Foursquare, our method higher than existing methods than 4.56%, 6.12%, 3.41% in terms of Acc@1, Acc@5, F1.

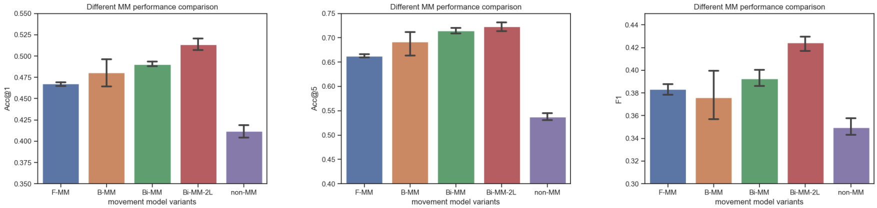
Table 3. Ablation experiment model settings

Model variants	Description
F-MM	Location prediction-based movement model with forward prediction only
B-MM	Location prediction-based movement model with back prediction only
Bi-MM	Location prediction-based movement model with bi-direction prediction
Bi-MM-2L	Location prediction-based movement model with 2-layer bi-direction prediction
Non-MM	Location embedding [8] without MM

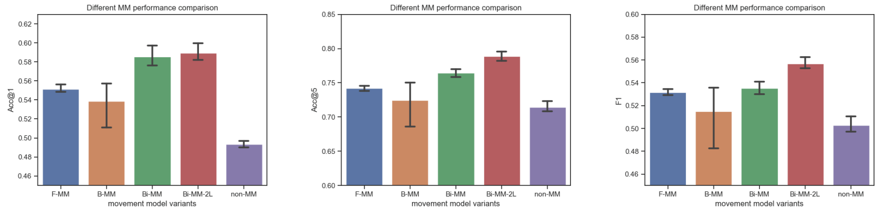
4.7 Case Study (Q3)

We select some specific trajectories from real-world datasets for analysis and visualization. First, we verify whether DBRLr can distinguish polysemous locations. We select a location[7] that is embedded to multiple vectors in different directions under DBRLr, and two trajectories contained the location mentioned above. In the single semantic location representation, these two trajectories are connected to unmatched users. While under DBRLr, T_1 and T_2 are respectively linked to the correct corresponding candidate users. We visualize the two trajectories and the above location l_3 shown in Fig. 5. By looking up in Google maps[8], we find that l_3 may be a bank, and T_1 passes through l_3 between two restaurants.

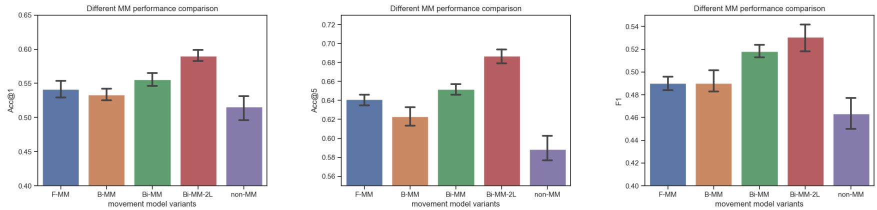
[7] Location ID is efa6e44dfa0145249be273ecd84a97f534b04920 in Brightkite.
[8] https://www.google.com/maps.



(a) Gowalla



(b) Brightkite



(c) Foursquare

Fig. 4. Performance comparison on different movement models.

We speculate that the user is handling business at l_3. On the contrary, T_2 passes another band and stays in l_3 for a long time, and we speculate that the user is working at l_3.

Secondly, we verify whether DBRLr contributes to the improvement of TUL recognition accuracy. We input four users' trajectories into the Linker under different location representations and exact the output layer for visualization, which is shown in Fig. 6. We notice that under DBRLr, the trajectories of the same users are more compact and clustered. However, in a single representation, the clusters are more scattered. This indicates that outliers decreases, while the clustered points increases under DBRLr, which is a practical demonstration that DBRLr can overcome misunderstandings of trajectory semantic information when linking anonymous trajectories to candidate users.

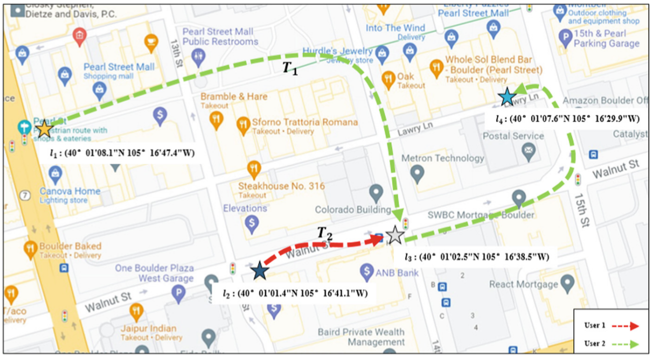
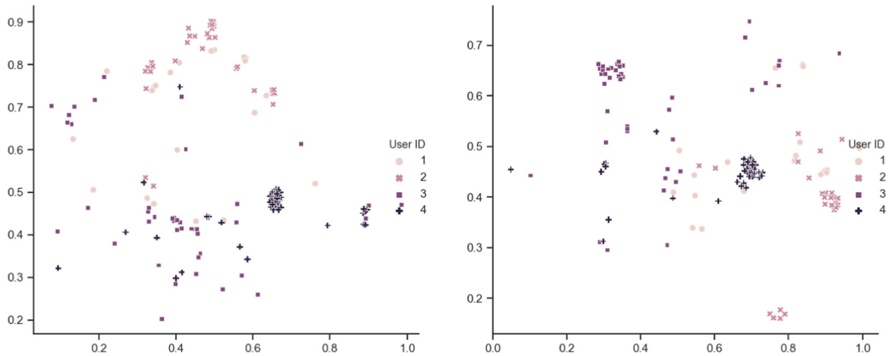


Fig. 5. Visualization of polysemous location in Brightkite.



(a) Single representation (b) DBRLr

Fig. 6. Trajectory embedding visualization.

5 Conclusions

In this paper, we improved TUL performance by using a polysemous location representation model called DBRLr. To learn this representation, we build a LP-based MM and train it historical trajectory corpora. Compared with the previous work, the trajectory with DBRLr can better describe the behavior and movement characteristics of a user. The experiment results confirm this view. We sincerely believe that DBRLr can be used not only in TUL but also in other trajectory data mining fields [25], such as next visited location prediction or location-based recommendation. Our follow-up work will examine our ideas on more datasets and more trajectory analysis tasks.

Acknowledgements. This work is partially supported by NSFC No. 61902376.

References

1. Chang, B., Park, Y., Park, D., Kim, S., Kang, J.: Content-aware hierarchical point-of-interest embedding model for successive poi recommendation. In: 27th IJCAI, vol. 2018 (2018)
2. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1082–1090 (2011)
3. Endo, Y., Toda, H., Nishida, K., Ikedo, J.: Classifying spatial trajectories using representation learning. Int. J. Data Sci. Anal. **2**(3), 107–117 (2016). https://doi.org/10.1007/s41060-016-0014-1
4. Feng, J., et al.: DeepMove: predicting human mobility with attentional recurrent networks. In: Proceedings of the 2018 World Wide Web Conference, pp. 1459–1468 (2018)
5. Feng, S., Cong, G., An, B., Chee, Y.M.: POI2Vec: geographical latent representation for predicting future visitors. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
6. Ferrero, C.A., Alvares, L.O., Zalewski, W., Bogorny, V.: MOVELETS: exploring relevant subtrajectories for robust trajectory classification. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing, pp. 849–856 (2018)
7. Gao, Q., Zhang, F., Yao, F., Li, A., Mei, L., Zhou, F.: Adversarial mobility learning for human trajectory classification. IEEE Access **8**, 20563–20576 (2020)
8. Gao, Q., Zhou, F., Zhang, K., Trajcevski, G., Luo, X., Zhang, F.: Identifying human mobility via trajectory embeddings. In: IJCAI, vol. 17, pp. 1689–1695 (2017)
9. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
10. Jiang, X., de Souza, E.N., Pesaranghader, A., Hu, B., Silver, D.L., Matwin, S.: TrajectoryNet: an embedded GPS trajectory representation for point-based classification using recurrent neural networks. arXiv preprint arXiv:1705.02636 (2017)
11. Liu, X., Liu, Y., Li, X.: Exploring the context of locations for personalized location recommendations. In: IJCAI, pp. 1188–1194 (2016)
12. Petry, L.M., Da Silva, C.L., Esuli, A., Renso, C., Bogorny, V.: MARC: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings. Int. J. Geog. Inf. Sci. **34**(7), 1428–1450 (2020)
13. Miao, C., Wang, J., Yu, H., Zhang, W., Qi, Y.: Trajectory-user linking with attentive recurrent network. In: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, pp. 878–886 (2020)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
15. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751 (2013)
16. Pang, J., Zhang, Y.: DeepCity: a feature learning framework for mining location check-ins. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 11 (2017)

17. Peters, M.E., et al.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
18. Qian, T., Wang, F., Xu, Y., Jiang, Yu., Sun, T., Yu, Y.: CABIN: a novel cooperative attention based location prediction network using internal-external trajectory dependencies. In: Farkaš, I., Masulli, P., Wermter, S. (eds.) ICANN 2020. LNCS, vol. 12397, pp. 521–532. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61616-8_42
19. Qian, T., Liu, B., Nguyen, Q.V.H., Yin, H.: Spatiotemporal representation learning for translation-based POI recommendation. ACM Trans. Inf. Syst. **37**(2), 1–24 (2019). https://doi.org/10.1145/3295499
20. Seglem, E., Züfle, A., Stutzki, J., Borutta, F., Faerman, E., Schubert, M.: On privacy in spatio-temporal data: user identification using microblog data. In: Gertz, M., et al. (eds.) SSTD 2017. LNCS, vol. 10411, pp. 43–61. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64367-0_3
21. Shimizu, T., Yabe, T., Tsubouchi, K.: Improving land use classification using human mobility-based hierarchical place embeddings. In: 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), pp. 305–311. IEEE (2021)
22. Sun, T., Xu, Y., Wang, F., Wu, L., Qian, T., Shao, Z.: Trajectory-user link with attention recurrent networks. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 4589–4596. IEEE (2021)
23. Wang, S., Cao, J., Yu, P.: Deep learning for spatio-temporal data mining: a survey. IEEE Trans. Knowl. Data Eng. (2020)
24. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based POI embedding for location-based recommendation. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 15–24 (2016)
25. Xu, Y., Liu, X., Cao, X., et al.: Artificial intelligence: a powerful paradigm for scientific research. Innovation **2**(4), 100179 (2021). https://doi.org/10.1016/j.xinn.2021.100179. https://www.sciencedirect.com/science/article/pii/S2666675821001041
26. Yabe, T., Tsubouchi, K., Shimizu, T., Sekimoto, Y., Ukkusuri, S.V.: City2City: translating place representations across cities. In: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 412–415 (2019)
27. Yan, B., Janowicz, K., Mai, G., Gao, S.: 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, pp. 1–10 (2017)
28. Yu, Y., et al.: TULSN: siamese network for trajectory-user linking. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2020)
29. Zhou, F., Gao, Q., Trajcevski, G., Zhang, K., Zhong, T., Zhang, F.: Trajectory-user linking via variational autoencoder. In: IJCAI, pp. 3212–3218 (2018)
30. Zhou, Y., Huang, Y.: DeepMove: learning place representations through large scale movement data. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 2403–2412. IEEE (2018)
31. Zhu, M., et al.: Location2vec: a situation-aware representation for visual exploration of urban locations. IEEE Trans. Intell. Transp. Syst. **20**(10), 3981–3990 (2019). https://doi.org/10.1109/tits.2019.2901117