Pre-training Context and Time Aware Location Embeddings from Spatial-Temporal Trajectories for User Next Location Prediction

Yan Lin,1,2 Huaiyu Wan,1,2,3 Shengnan Guo,1,2,4 Youfang Lin1,2,3,4*

1School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
2Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing, China
3CAAC Key Laboratory of Intelligent Passenger Service of Civil Aviation, Beijing, China
4Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport, Beijing, China
{ylincs, hywan, guoshn, yflin}@bjtu.edu.cn

Abstract

Pre-training location embeddings from spatial-temporal trajectories is a fundamental procedure and very beneficial for user next location prediction. In the real world, a location usually has variable functionalities under different contextual environments. If the exact functions of a location in the trajectory can be reflected in its embedding, the accuracy of user next location prediction should be improved. Yet, existing location embeddings pre-trained on trajectories are mostly based on distributed word representations, which mix a location’s various functionalities into one latent representation vector. To address this problem, we propose a Context and Time aware Location Embedding (CTLE) model, which calculates a location’s representation vector with consideration of its specific contextual neighbors in trajectories. In this way, the multi-functional properties of locations can be properly tackled. Furthermore, in order to incorporate temporal information in trajectories into location embeddings, we propose a subtle temporal encoding module and a novel pre-training objective, which further improve the quality of location embeddings. We evaluate our proposed model on two real-world mobile user trajectory datasets. The experimental results demonstrate that, compared with the existing embedding methods, our CTLE model can pre-train higher quality location embeddings and significantly improve the performance of downstream user location prediction models.

Introduction

In recent years, the increasing availability of location-based service (LBS) data, such as GPS trajectories, check-ins at point-of-interests (POIs) and cellular signaling records, have led to a burst of studies focused on mining spatial-temporal data. Among them, user location prediction has received much attention (Liu et al. 2016; Xiao et al. 2017; Kong and Wu 2018), and can be utilized in multiple applications, including modeling users’ mobility behaviors (Fu and Lee 2020), or recommending locations (POIs) for users (Zhao et al. 2019). Among these researches, learning embedding vectors for locations is a very fundamental and critical problem, due to the fact that accurate user location prediction requires high quality embedding vectors for better discrimination between locations. Compared to the widely used fully-connected embedding layers in most prediction models, pre-training location embeddings using un-supervised or self-supervised objectives can incorporate more general and comprehensive information of locations. In this way, they can be shared across a wide range of downstream models to improve the computation efficiency, while also promoting the overall prediction performance.

The idea of capturing location characteristics from sequential trajectories is akin to language modelings in natural language processing (NLP). In fact, most existing pre-trained location embedding methods are based on distributed word representations, which are widely used in NLP systems. For example, DeepMove (Zhou and Huang 2018) implements Word2Vec (Mikolov et al. 2013) on trajectory data to extract location function information. However, besides sharing many similar characteristics with sentences, trajectories contain unique temporal information that shouldn’t be ignored. As illustrated in Figure 1(a), in a trajectory, the relative visited time differences between locations can reflect their properties such as visited frequency and stayed duration. On the other hand, time will affect peoples’ preferences on locations, so the visited time of a location can be an indication of its functionalities, as shown in Figure 1(b). Geo-
neighbors. In this paper, we propose a novel
embedding vectors for locations based on their specific contextual
training model which is able to adaptively generate embedding
location embedding models, we aim to build a pre-
distributed representations assign a single latent vector for
accurate. Yet, current location embedding methods based on
different purposes in dissimilar contextual neighbors, as
shown in Figure 2. Incorporating the specific context of a
location into its embedding vector can be beneficial for lo-
cation prediction, for the provided information will be more
yet, current location embedding methods based on
distributed representations assign a single latent vector for
each location. This approach can be viewed as a simple mix-
ing of various contextual environments, and is unable to han-
dle variable location functionalities under different contexts.

In order to tackle the above discussed problems lie in ex-
isting location embedding models, we aim to build a pre-
training model which is able to adaptively generate embed-
ding vectors for locations based on their specific contextual
neighbors. In this paper, we propose a novel Context and
Time aware Location Embedding (CTLE) model. Instead
of assigning one latent vector as each location’s final rep-
resentation, a target location’s embedding vector is calcu-
lated by a parameterized mapping function of its contextual
locations’ encoding vectors. In this way, we are able to
incorporate locations’ context-specific functionalities into
their embedding vectors. We implement the mapping func-
tion using bidirectional Transformer encoder (Vaswani et al.
2017), and employ Masked Language Model (MLM) pre-
training objective introduced in BERT (Devlin et al. 2019)
to model the sequential correlations in user trajectories.
Furthermore, in order to incorporate the unique temporal in-
formation in trajectories, we design a subtle temporal encod-
ing module to model relative visited time differences be-
 tween locations. And we also propose a novel Masked Hour
(MH) pre-training objective to extract locations’ functional-
ities from their visited time distributions. Our model’s effec-
tiveness is verified on two real-world mobile user trajectory
datasets and three downstream models for user next loca-
tion prediction. The experimental results demonstrate that
our CTLE model can obviously help downstream models to
improve the prediction accuracy on dense trajectories.

The main contributions of this paper are summarized as
follows:
• We propose a contextual location embedding model
  CTLE, which is able to adaptively generate embedding
  vector for a target location based on its specific context.
  Therefore, a location’s variable functionalities under dif-
  ferent contextual environments can be discriminated.
• A subtle temporal encoding module and a novel pre-
  training objective are proposed to model two aspects of
temporal information in trajectories. This will improve
our model’s understanding of locations’ characteristics.
• We employ our CTLE model into three downstream mod-
els for user next location prediction, and conduct exten-
sive experiments on two real-world mobile user trajec-
tory datasets. The experimental results show that the pre-
diction performances are significantly improved, which
demonstrates our model’s superiority.

Related Work

Most user location prediction models are feature based, and
require locations being represented by latent embedding
vectors. One of the most straightforward and widely applied
strategies is to use a fully-connected embedding layer. This
layer randomly initializes one latent vector for each location,
which is then trained with task-specific objectives (Kong and
Wu 2018; Zhao et al. 2019). Yet, the embedding vectors gen-
erated in this way are hard to migrate to other models and
tasks. Embedding layers also suffer from over-fitting prob-
lems, and are unable to incorporate comprehensive informa-
tion of locations, which will hurt prediction performance.

Pre-training embedding vectors with un-supervised or
self-supervised objectives is a common practice in natu-
ral language processing (Mikolov et al. 2013; Devlin et al.
2019) and computer vision (Szegedy et al. 2015). In recent
years, this topic has also attracted much attention in spatial-
temporal data mining. Actually, as a kind of sequential data,
user mobility trajectories share many common characteristics
with sentences in natural language. Thus, by modeling
the co-occurrence of target locations and their contexts in
 trajectories, we can extract locations’ functionality informa-
tion. Inspired by this idea, most existing location embed-
ning models are based on word embedding models in natu-
ral language processing. For example, DeepMove (Zhou
and Huang 2018) implements Skip-gram (Mikolov et al.
2013) to model human movements between locations, and
(Yao et al. 2018) learns latent representations through an
N-gram model (Pauls and Klein 2011). These pre-training
models only require un-labeled trajectory data, which are
often abundant. The quality of result embedding vectors are
validated by incorporating them into downstream tasks, like
user location prediction (Zhao et al. 2017) and land usage
classification (Zhang et al. 2020).

However, user trajectories also include some unique prop-
erties, and temporal correlation is one of them. Locations’
functionalities can be more accurately extracted from the
temporal information, so it should be considered by embed-
ding models. Geo-teaser (Zhao et al. 2017) discriminates
locations which are visited during weekdays and week-
ends through expanding the output embeddings in Skip-
gram. TALE (Wan et al. 2019) incorporates visited time
information by designing a novel tree structure for soft-
max calculation in CBOW (Mikolov et al. 2013). HIER
(Shimizu, Yabe, and Tsubouchi 2020) combines visited time and stayed duration information by feeding an RNN-based N-gram model with extra embedding vectors. Experimental results show temporal information is indeed helpful for pre-training higher quality location embedding vectors.

It is common that a location is multi-functional, and people may visit the same location for various purposes under dissimilar contextual environments. Correspondingly, it is possible to indicate a location’s specific functionality from its context, i.e., nearby visited locations in the trajectory. Yet, existing location embedding methods only assign a single latent vector for each location, which means they are unable to differentiate a location’s variable functionalities. Inspired by the recent advances of contextual embedding in natural language processing (Devlin et al. 2019; Yang et al. 2019), we propose a context-aware location embedding model that dynamically calculate latent embeddings for locations based on their contextual neighbors. In this way, the specific functionality of a location is able to be well modeled.

Preliminaries

Definition. Spatial-Temporal Trajectory. In location-based services, a user’s movements during a certain period can be represented by a trajectory $s$ consisting of sequential visiting records. A visiting record $(u, l, t)$ indicates that user $u$ visited location $l$ at time $t$. We denote the set of user trajectories as $S$, the set of all locations appear in the dataset as $L$, and the set of all users as $U$.

Problem Statement. Pre-training Contextual Embedding Vectors for Locations. Given a set of user spatial-temporal trajectories $S$, we aim to pre-train a parameterized mapping function $f$ to generate a contextual embedding vector $z(l)$ for a target location $l$ given its context $C(l)$. The function should be trained in a self-supervised manner with no task-specific objectives needed.

The CTLE Model

Figure 3 illustrates the architecture of our CTLE model. It mainly consists of three components: 1) the encoding layer, which fuses temporal encoding vectors with locations’ encoding vectors to extract information from relative visited time differences; 2) the bidirectional Transformer encoder, acting as a mapping function for calculating a target location’s embedding, given its specific contextual neighbors; 3) the pre-training objective, which incorporates locations’ characteristic information into their embeddings through modeling the co-occurrence probability between target locations and contexts, and the absolute visited time to the targets. The construction of our CTLE model is explained in detail in this section.

Pre-training Bidirectional Transformer for Context-aware Location Embedding

Multi-functional locations are very common in the real world. Usually, a user visits one location with a certain purpose, but if the location is multi-functional, a single visiting record $(u, l, t)$ cannot directly indicate the location’s functionality under that specific circumstance. Contextual environment, i.e., visiting records nearby the target location in a trajectory, can more explicitly indicate a certain functionality of the target location, as shown in Figure 2.

Based on the above observation, we propose a context-aware location embedding model, which generates a location’s latent representation with consideration of its contextual neighbors. Specifically, given a target location $l$ and its context $C(l)$, we calculate its embedding vector $z(l)$ via a parameterized mapping function $f$, denoted as the following abstract form:

$$z(l) = f(l, C(l)).$$

In this way, locations’ embedding vectors are dependent on their contextual environments, which makes it possible to discriminate a location’s specific functionality. To implement function $f$, we utilize a bidirectional Transformer encoder (Vaswani et al. 2017), for two reasons. First, locations in trajectories are related with contexts from both left and right sides. Bidirectional Transformers are able to capture contextual information from both sides, while also considering their interactions. Second, Transformer architecture has been proven by many studies (Dai et al. 2019) to have higher expressive power than traditional sequential models like LSTM (Hochreiter and Schmidhuber 1997).

Now we introduce the calculation of contextual embeddings in detail. Given a trajectory $s = \{(u, l_1, t_1), (u, l_2, t_2), \ldots, (u, l_n, t_n)\}$, we fetch an input latent vector $z'(l)$ for each location $l$ from the location encoding module, denoted as:

$$z'(l) = \Omega(l),$$

thus forming an input sequence $\{z'(l_1), z'(l_2), \ldots, z'(l_n)\}$. $\Omega$ denotes the location encoding module, which is implemented using a fully-connected embedding layer. The input sequence is then fed into the stack of Transformer encoder layers, where each layer is composed of a multi-head self-attention module and a position-wise fully connected feedforward network. Formally, we have:

$$\{h_1^{(k)}, h_2^{(k)}, \ldots, h_n^{(k)}\} = \text{TransEncLayer}(\{h_1^{(k-1)}, h_2^{(k-1)}, \ldots, h_n^{(k-1)}\}),$$

where $\text{TransEncLayer}$ represents a Transformer encoder layer. $\{h_1^{(k)}, h_2^{(k)}, \ldots, h_n^{(k)}\}$ is the output memory sequence of the $k$-th layer, and the input sequence of the $(k + 1)$-th layer. Suppose the model contains a total of $N$ Transformer encoder layers. We regard the $i$-th item of the last layer’s output memory sequence, $h_i^{(N)}$, as the contextual embedding vector of location $l_i$ in trajectory $s$. If we denote all locations in $s$ except $l_i$ as its context $C(l_i)$, the calculation equation of $z(l_i)$ can be written as:

$$z(l_i) = h_i^{(N)} = \text{TransEnc}(\{z'(l_1), z'(l_2), \ldots, z'(l_n)\}_i) = \text{TransEnc}(\Omega(l_1, l_2, \ldots, l_n))_i = f(l_1, l_2, \ldots, l_n)_i = f(l_i, C(l_i))_i,$$
which corresponds to the abstract form presented in Equation (1). TransEnc represents the stack of Transformer encoder layers.

The mapping function $f$ should learn to understand the correlation between target locations and their corresponding contexts, so that the locations’ characteristic information can be incorporated into their embedding vectors. Inspired by recent advances of language understanding in NLP models, we implement the Masked Language Model (MLM) proposed in BERT (Devlin et al. 2019) to construct a self-supervised training objective. Given a trajectory $s$, we randomly select 20% records as masked records, and replace their corresponding locations and visited time with special tokens $[m_n]$ and $[m_n]$. The result trajectory $\tilde{s}$ is regarded as the input of the function $f$. For a masked location $l_m$, we aim to predict its original token using the output of $f$:

$$\hat{l}_m = \text{FFN}_{\text{MLM}}(\text{TransEnc}(\tilde{s})) = \text{FFN}_{\text{MLM}}(f(\tilde{s})),$$

where $\text{FFN}_{\text{MLM}}$ is a fully-connected feed-forward network for casting Transformer encoder’s output memory vectors into the prediction of location tokens. MLM objective can then be constructed through maximizing the prediction accuracy. If we regard the masked locations as targets, and the un-masked locations as their contexts, the pre-training objective of MLM can be described as:

$$O_{\text{MLM}} = \arg \max_\theta \sum_{l_m \in \Gamma} \text{FFN}_{\text{MLM}}(f(\tilde{s}))$$

where $\Gamma$ denotes the set of all masked records in the training set, $\theta$ denotes the set of all learnable parameters of the embedding model. By maximizing the co-occurrence probability of targets and their contexts, locations with similar contextual environment will have closer embedding vectors, thus the functionality information is incorporated. This idea is akin to that in Word2Vec (Mikolov et al. 2013). Yet, instead of directly optimizing a set of static embedding vectors, our model optimizes the parameters of a mapping function. In this way, the result embedding vectors will be context-dependent, and are able to incorporate locations’ specific characteristic information in different contextual neighbors.

**Incorporating Temporal Information**

Temporal information is essential for spatial-temporal data mining. For trajectory data, temporal information is mainly carried by the visited time of locations, and can be viewed from two aspects: relative visited time difference and absolute visited time. Locations’ visited frequencies or stayed duration can be revealed from relative visited time differences, which implies functionality information, as illustrated by Figure 1(a). On the other hand, human activities are usually time-regulated, and the functionalities a location carried can determine the time users arrive at it, as shown in Figure 1(b). Correspondingly, we can extract characteristic information from locations’ absolute visited time.

Based on the above analysis, we incorporate two aspects of temporal information into our model via a subtle temporal encoding module and a novel pre-training objective.

**Temporal Encoding in Encoding Layer** In order to explicitly model the relative visited time differences between locations in trajectories, we propose the temporal encoding module. In the original Transformer (Vaswani et al. 2017), positional encoding is introduced for the model to discriminate the order of tokens in the input sequence. Its calculation equation can be written as:

$$\Phi(c) = [\cos(\omega_1 o), \sin(\omega_1 o), \ldots, \cos(\omega_d o), \sin(\omega_d o)],$$

$$\omega_k = \frac{1}{10000^{2k/d}},$$

where function $\Phi$ represents the positional encoding module, $o$ is the index of a position, and $2d$ is the dimension of positional encoding vector.

Positional encoding can only represent simple sequential orders. Yet, for trajectories, the visiting records unevenly ar-
range on the temporal axis, and the relative visited time differences between locations contain important information, like locations’ visited frequencies or stayed durations, as shown in Figure 1(a). Inspired by (Xu et al. 2020), our proposed temporal encoding tackles this problem by making two simple adjustments to the positional encoding: replacing the index of position \(o\) with an absolute visited timestamp \(t\), and setting the parameters \(\{\omega_1, \omega_2, \ldots, \omega_d\}\) to be trainable. Formally, we have:

\[
\Psi(t) = [\cos(\omega_1 t), \sin(\omega_1 t), \ldots, \cos(\omega_d t), \sin(\omega_d t)],
\]

where \(\Psi\) is our temporal encoding module. The dot-product of \(\Psi(t)\) and \(\Psi(t + \delta)\) are calculated as:

\[
\Psi(t) \cdot \Psi(t + \delta) = \cos(\omega_1 \delta) + \cos(\omega_2 \delta) + \cdots + \cos(\omega_d \delta),
\]

which means the distance (measured by dot-product) between \(\Psi(t)\) and \(\Psi(t + \delta)\) is learnable, and is independent of \(t\). There are no lack of dot-product operations in a Transformer encoder, so this distance information is destined to be captured by the model.

Then, we introduce the temporal encoding into our model by combing it with the location encoding. Specifically, we modify Equation 2 for fetching the input latent vector of record \((u, l, t)\) as follows:

\[
z'(l) = \Omega(l) + \Psi(t).
\]

In this way, our model is able to incorporate the temporal information embedded in relative visited time differences into the pre-training and embedding generation process, which results in higher quality embedding vectors. The overall architecture of CTLE’s encoding layer combined with temporal encoding is shown in Figure 3(a).

**Masked Hour Objective in Pre-training** In order to further extract information from the absolute visited time, we propose the Masked Hour (MH) pre-training objective. For a masked visited time \(t_m\) in a masked trajectory \(\tilde{s}\), we aim to predict its original visited hour index:

\[
\text{Hour}(t_m) = \text{FFN}_{\text{MH}}(f(\tilde{s})_m),
\]

where \(\text{Hour}(t)\) denotes the hour index of time \(t\). \(\text{FFN}_{\text{MH}}\) is a fully-connected network for casting the Transformer encoder’s output memory vector into the prediction of hour index. During pre-training, the MH is combined with MLM to form a multi-task learning objective. The MH objective and the overall pre-training objective can be described as:

\[
O_{\text{MH}} = \arg \max_\theta \sum_{l_m \in \mathcal{C}} \Pr(\text{Hour}(t_m) | \text{FFN}_{\text{MH}}(f(\tilde{s})_m)),
\]

\[
O = O_{\text{MLM}} + O_{\text{MH}},
\]

where \(\theta\) denotes the set of all learnable parameters of embedding model. We can give an abstract frame of the above pre-training objective as \(\arg \max_\theta p(l_m, t_m | C(t_m))\). By maximizing the probability that a target location will be visited during a certain period given its contexts, we incorporate the information implied in absolute visited time into the model. This enables the model’s ability to extract locations’ functionalities from temporal information, thus improving the quality of generated embedding vectors. The overall model architecture of CTLE during the pre-training phase is shown in Figure 3(b).

**Scope of Application**

To make the contextual embedding idea of CTLE effective, the trajectories need to be relatively dense, so that target locations have adequate strength of correlations with their contextual environments. In other words, the contexts need to be complete enough to reveal the targets’ functionalities. That is what sparse trajectories such as most of check-in records lack. Besides, locations in check-in data are often manually divided and classified into POIs with certain purposes, which means their multi-functional property is not obvious. In summary, when applied on very sparse trajectories like check-in data, our CTLE will lose its superiority over the existing distributed embedding-based methods.

**Experiments**

In order to evaluate the quality of contextual embedding vectors generated by our model, we incorporate these vectors into three downstream user location prediction models, and compare the results with other location embedding methods.

**Datasets**

The experiments are carried out on two real-world datasets which contain mobile signaling data in Beijing and Shenyang in China, denoted as Mobile-PEK and Mobile-SHE, respectively. They record the switching events between telecommunication base stations of anonymous mobile users. Each base station is responsible for providing signal to its surrounding area, and users’ entering to the area will be recorded by the station. In this paper, we treat base stations as locations. Unlike some public available check-in datasets, mobile signaling datasets contain dense trajectories, making them more proper for evaluating the effectiveness of CTLE.

Technically, signaling data can be representative of a user’s full moving trajectories. Yet, not all trajectory points reflect the user’s visiting to locations. For example, people’s traveling from one place to another will also generate a sequence of trajectory points, but these points lack semantic information. In order to filter out such points that users just passed by, we firstly remove the base stations and users whose related records have an average duration time below 30 minutes, for most signaling records generated by them are just passing-points. Then we remove all the trajectory points with a duration time below 5 minutes. After this processing, all remaining signaling records can be seen as staying points. A user’s staying point sequence within one day sorted by ascending timestamp is regarded as a trajectory. Finally, we include the trajectories with more than 5 staying points and users with more than 4 trajectories in our datasets. The statistics of datasets are shown in Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Users</th>
<th>#Locations</th>
<th>#Records</th>
<th>Time span</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile-PEK</td>
<td>12,691</td>
<td>7,279</td>
<td>1,383,422</td>
<td>5 days</td>
</tr>
<tr>
<td>Mobile-SHE</td>
<td>10,564</td>
<td>7,201</td>
<td>607,581</td>
<td>11 days</td>
</tr>
</tbody>
</table>

Table 1: Statistics of datasets.
Baseline Location Embedding Methods

To prove the superiority of our model, we include two classic distributed embedding models which are implemented by many location embedding methods, and also some state-of-the-art location embedding models for comparison.

- **FC Layer**: a widely used strategy for latent representation, which allocates a randomly initialized embedding vector for each location. These vectors are then trained together with other parameters of the whole models.
- **Skip-gram** (Mikolov et al. 2013): a variant of the Word2Vec model, and has been utilized in (Liu, Liu, and Li 2016) for modeling mobility trajectories.
- **POI2Vec** (Feng et al. 2017): an embedding method based on Word2Vec, which models spatial correlations through distributing locations into a geographical binary tree.
- **Geo-Teaser** (Zhao et al. 2017): Geo-Temporal Sequential Embedding Rank fuses temporal and spatial information through expanding the output vector, and modifying Skip-gram's negative sampling strategy.
- **TALE** (Wan et al. 2019): Time-Aware Location Embedding incorporates temporal information through designing a novel temporal tree structure for hierarchical softmax calculation.

User Next Location Prediction Models

Downstream prediction models can be utilized to evaluate the effectiveness of embedding methods, for the prediction performance relies on the quality of location embedding vectors. In this paper, we employ three popular user next location prediction methods as the downstream models:

- **ERPP** (Xiao et al. 2017): Event Recurrent Point Process regards temporal event sequences as point processes. In this paper, we consider a visit to a location as an event, and the arrival time as its timestamp.
- **ST-LSTM** (Kong and Wu 2018): A Spatial-Temporal Long-Short Term Memory model for user next location prediction. It expands the input of LSTM by embedding time and distance values into latent vectors.

Give a sequential trajectory \(s = \{(u_{11}, t_{11}), (u_{12}, t_{12}), \ldots, (u_{tn}, t_{tn})\}\), we feed the prediction models with a sequence of corresponding embedding vectors \(\{z(\ell_1), z(\ell_2), \ldots, z(\ell_t)\}\). It’s worth noting that when applying CTLE to the location prediction models, we use a special attention mask to prevent the information of the locations to be predicted from leaking to the historical sequence, as illustrated in Figure 3(c). All embedding methods except FC Layer do not participate in backward propagation during location prediction.

Settings

For the Mobile-PEK dataset, we choose trajectories in the first 3 days for training, the 4th day for evaluation and the last day for test. For the Mobile-SHE dataset, we choose trajectories in the first 7 days for training, the last 2 days for test and the other days for evaluation. In the location prediction task, we select the last three points in every trajectory as prediction targets, and the rest of points as source sequence.

Both pre-trained location embedding models and downstream prediction models are trained with the training sets, and validated on the evaluation sets to get the optimum hyperparameters. Note that the pre-training embedding methods themselves don’t have a reliable metric for the performance evaluation, so we adjust the hyperparameters with the help of the ST-LSTM model. The location prediction models are trained with Cross Entropy loss, and evaluated with Accuracy, macro-Recall and macro-F1 score.

We implement all baseline models and our CTLE model in PyTorch (Paszke et al. 2019). All embedding models’ dimensions of result embedding vectors are set to 128. We implement ERPP and ST-LSTM based on 2-layer LSTM networks with a hidden size of 512. The unique spatial-temporal context structure of ST-RNN prevents us from implementing it in a multi-layer manner, so it is based on a single-layer RNN with a hidden size of 512. For our CTLE model, we stack 4 layers of Transformer encoders, with 8 attention heads, and the hidden sizes of feed-forward layers are set to 512. The CTLE is pre-trained on the training sets for 200 epochs, and all prediction models are trained with the early-stopping mechanism to obtain the best-performing epochs on the evaluation sets. We choose Adam optimizer and an initial learning rate of 0.0001 across the board. All experiments were ran on Intel Xeon Silver 4210 CPUs, and NVIDIA RTX2080ti GPUs.

Experimental Results

Table 2 shows the performance comparison of different models for user next location prediction. All baseline embedding methods are fused with multiple downstream prediction models to calculate the results. Our CTLE consistently outperforms other location embedding methods except on the ST-RNN prediction model, Mobile-SHE dataset. Randomly initialized fully-connected embedding layers only fit on the specific prediction objectives, and are unable to incorporate general information, like the functionalities of locations. Skip-gram only utilizes the co-occurrence probabilities of target locations and their contexts, ignoring some unique properties of spatial-temporal trajectories, like the visited time and geographical positions of locations. As we can see from Table 2, these methods generally result in worse prediction performance than the newer, more comprehensive location embedding methods.

Geo-Teaser and POI2Vec introduce spatial information into the embeddings through different approaches. Yet, geographical correlation between locations in a city is not very evident. On the one hand, people’s visiting preferences are not confined within small areas in the city; on the other hand, locations’ properties don’t have a strong correlation with their geographical positions, as close-up locations might
also have diverse functionalities.

Geo-Teaser also incorporates temporal information into location embeddings, by taking absolute visited time of locations into consideration. Yet, it only discriminates visiting records happened during weekdays and weekends, ignoring finer-grained temporal information, like visiting hours. TALE and HIER both utilize visiting hour indices, which can imply users’ preferences to locations. Thus, they can generate location embeddings that are more beneficial for prediction tasks. Yet, none of the aforementioned location embedding methods considers relative visited time differences, which are capable of indicating relationship between locations. More importantly, they assign only one latent vector to represent each location, which makes them unable to reflect the variable functionality property of locations in resulting embedding vectors. This can seriously hurt the prediction performance of downstream models. Unlike check-in datasets, locations in mobile signaling data are not manually divided and classified into POIs with certain purposes, so their multi-functional property is extra obvious.

Our CTLE model dynamically generates locations’ embeddings with regards to their specific context, so that the exact functionality of a location in a certain contextual environment can be discriminated. In addition, CTLE takes both relative visited time differences between locations and fine-grained absolute visited time into consideration. As demonstrated in Table 2, these designs result in higher quality location embeddings, and can help downstream location prediction models to achieve better performance.

**Table 2: User next location prediction performance comparison of different approaches.**

![Figure 4: Component analysis of CTLE.](image_url)

**Component Analysis**

To further investigate the effectiveness of each component of CTLE, we design three variants of the CTLE model, as listed below:

- Basic: This model uses the original positional encoding from Transformer (Vaswani et al. 2017), and only utilizes the MLM objective for pre-training.
- +TempEnc: This model replaces the positional encoding in the basic model with our proposed temporal encoding.
- +MH: This model still uses positional encoding, but combines the Masked Hour objective with the MLM objective during pre-training.

We compare these three variants with the CTLE model on the ST-LSTM downstream model. Figure 4 shows the results. The basic model already have some performance advantages compared to the location embedding methods that are based on distributed word embeddings. This indicates that learning context-aware location embedding vectors can be beneficial for the location prediction task.

Temporal encoding and the Masked Hour objective incorporates temporal information embedded in trajectories into the model. As illustrated in Figure 4, they can both improve the prediction performance over the basic model. Because these two modules focus on different aspects of temporal influence, our final model combines them and gain the optimum results.

**Conclusion**

In this paper, we propose a novel context and time aware location embedding method CTLE. It dynamically generates embedding vectors of target locations given their contexts, so the specific functionality of a location under certain contextual environment can be incorporated. A subtle temporal encoding module and a novel Masked Hour pre-training objective further incorporate multiple aspects of temporal information in trajectories into the embeddings. Experiments are conducted on the user next location prediction downstream task on two real-world mobile user trajectory datasets and three popular downstream models, and the results prove the effectiveness of our model for learning accurate and comprehensive location embeddings on dense trajectories.
Acknowledgments
This work was supported by the Fundamental Research Funds for the Central Universities (Grant No. 2019JBM024).

References


