CASTLE: A CONTEXT-AWARE SPATIAL-TEMPORAL LOCATION EMBEDDING PRE-TRAINING MODEL FOR NEXT LOCATION PREDICTION

Junyi Cheng [1], Jie Huang [1], Xianfeng Zhang [1] *

[1] Institute of Remote Sensing and Geographic Information Systems, Peking University, 5 Summer Palace Road, Beijing, 100871, China - (junyicheng, jiehuang, xfzhang)@pku.edu.cn

KEY WORDS: Geospatial Data, Ubiquitous Computing, Location Prediction, Location Embedding, Trajectory Mining, Smart City.

ABSTRACT:

Next location prediction is helpful for service recommendation, public safety, intelligent transportation, and other location-based applications. Existing location prediction methods usually use sparse check-in trajectories and require massive historical data to capture complex spatial-temporal correlations. High spatial-temporal resolution trajectories have rich information. However, obtaining personal trajectories with long time series and high spatiotemporal resolution usually proves challenging. Herein, this paper proposes a two-stage Context-Aware Spatial-Temporal Location Embedding (CASTLE) model, a multi-modal pre-training model for sequence-to-sequence prediction tasks. The method is built in two steps. First, large-scale location datasets, which are sparse but easier to be acquired (i.e., check-in and anomalous navigation data), are used for pre-training location embedding to capture the multi-functional properties under different contexts. After that, the learned contextual embedding is used for downstream location prediction in small-scale but higher spatiotemporal resolution trajectory datasets. Specifically, the CASTLE model combines Bidirectional and Auto-Regressive Transformers to generate contextual embedding vectors rather than a fixed vector for each location. Furthermore, we introduce a location and time-aware encoder to reflect the spatial distances between locations and visit times. Experiments are conducted on two real trajectory datasets. The results show that the CASTLE model can pre-train beneficial location embedding and outperforms the model without pre-training by 4.6-7.1%. The proposed method is expected to improve the next location prediction accuracy without massive historical data, which will greatly drive the use of trajectory data.

1. INTRODUCTION

Next location prediction has raised intensive studies in recent years owing to the growth of location-based services. The large volume of historical data makes it possible to understand individuals' preferences for the next movements (Wan et al., 2021), as the trajectory data reveals individuals' travel patterns and preferences. Meanwhile, predicting the next location is of great significance for service recommendation, public safety, intelligent transportation, and other location-based applications (Luo et al., 2021).

There have been various models to predict the next location based on the historical trajectory in the past two decades. In general, location prediction methods can be categorized as pattern-based, probability distribution-based, statistical learning-based, and representation-learning-based. The pattern-based methods refer to extracting spatiotemporal patterns from historical trajectories for location prediction. Commonly used patterns include sequential, frequent, periodic, and clustering patterns. For example, the commonly used sequence mining model T-pattern tree records the behavior and the visit time of each location and calculates the transition probabilities between locations to dynamically predict the next location by finding the optimal matching path (Monreale et al., 2009). Based on historical trajectory data, some studies mine frequently visited locations of individuals through clustering and established a path network to predict the next location that individuals will go to (Yuan et al., 2014). However, it is not easy to extract a long-term effective and meaningful movement pattern, and the fixed pattern limits the diversity of model prediction results.

The core idea of the probability distribution-based method is to fit a probability calculation model to evaluate the user's visiting preferences for different locations and then predict the next visiting location. This method does not require parameter learning and training but only needs to fit the parameters according to a predetermined probability model based on historical trajectories. Specifically, this method establishes a probability calculation model from different aspects (e.g., geographic location, time, sequence characteristics, location category) based on the existing model (Zhang et al., 2015). An interesting study uses the Gaussian Mixture Model to fit the two-dimensional spatial distribution of the user's historically visited points of interest (Zhang and Chow, 2014). In general, the method based on probability distribution has good interpretability. However, methods based on probability distributions rely on prior knowledge, and fitting with different statistical models may yield different results.

The method based on statistical learning refers to obtaining the optimal parameter combination by training based on historical data, mainly including matrix factorization, topic, and other classification-based machine learning models. The basic idea of the matrix factorization models, e.g., RCH (Wang et al., 2015) and GeoMF (Lian et al., 2018), are to decompose the user-location matrix into two low-rank matrices representing the user and the location, respectively. In the subsequent research, the matrix dimension is expanded into a tensor, expressed as a user-time-location tensor. The tensor factorization method is used to analyze the temporal patterns of users' travel behavior (Bhargava et al., 2015). However, this kind of method is not suitable for cold-start problems, especially for new users and new locations

* Corresponding author

requiring the model's retraining. Meanwhile, it ignores the sequential correlations in the trajectory.

With the advancement of deep learning technology, representation learning has become widespread in next location prediction research. The core idea is to represent each location with a vector and train the model to get a latent embedding vector with a specific task. Similar to natural language, the trajectory is also a sequence in which the sequence node strongly correlates with its context. Thus, word embedding models in natural language processing have been widely used in the representation learning of trajectory. For example, the DeepMove model (Feng et al., 2018) uses the Skip-gram model to extract contextual information; that is, predicting the surrounding context through the central node. The Tale model (Wan et al., 2019) is based on the CBOW model to capture the temporal dependencies in the trajectory; that is, predicting the central nodes by the surrounding context. Although these methods can generate beneficial embedding vectors, they will produce a fixed embedding vector for the same location under a different context.

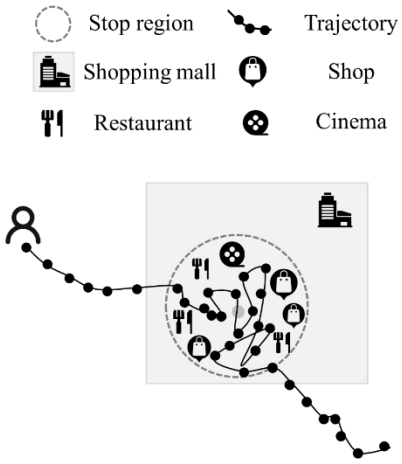

Figure 1. Uncertain visited places in real trajectories. One stop may match multiple different places.

Unlike the usually used check-in data, the visited location is uncertain in the real GPS or mobile phone trajectory dataset (Figure 1) due to the low spatial accuracy of the positioning terminal and the place ambiguity (e.g., multiple shop malls or cinemas in a shopping mall). It means that people may visit the same location for a different purpose; that is, a location may be multi-functional in the real world. Thus a model which can dynamically generate the contextual embedding vector is urgently needed. A recently proposed Bert-based (Devlin et al., 2019) CTLE (Lin et al., 2021) model uses a bidirectional encoder to generate the embedding for a location based on its spatial-temporal context. It shows that the dynamic location embedding significantly improves the downstream task's performance. However, it ignores the spatial proximity of the locations.

However, there still exist some problems in the existing methods. First, most existing studies use sparse check-in trajectory data, which is easy to acquire. However, the trajectory data obtained by mobile phones or GPS terminals with the high spatial-temporal resolution has rich information. Obtaining personal trajectories with long time series and high spatiotemporal resolution usually proves challenging. Furthermore, training an effective model will be a difficult task without massive historical data. Second, the trajectory contains rich spatial-temporal information, and the existing methods fail to capture spatiotemporal associations between visited locations effectively.

To address the above problems, we propose a two-stage context-aware spatial-temporal location embedding pre-training model for the next location prediction. The contribution can be summarized as follows:

(1) A two-stage framework is proposed to solve the problem that obtaining large-scale trajectories with the high spatial-temporal resolution is challenging. Thus, our model could predict the visit places accurately using small-scale fine-grained trajectory data.

(2) We propose an encoding layer that incorporates the spatial position and the temporal information. Therefore, preferences for travel distance and visit time can be reflected in the model.

(3) The bidirectional encoder and the autoregressive decoder are combined to dynamically capture long-term sequential dependence, which is more suitable for the uncertain visit places of real GPS or mobile phone trajectory.

2. METHODOLOGY

2.1 Proposed Framework for Next Location Prediction

Obtaining personal trajectories with long time series and high spatiotemporal resolution usually proves to be challenging. Thus, we propose a two-stage framework (Figure 2) for next location prediction, including pre-training contextual embedding vectors for locations and fine-tuned next location prediction.

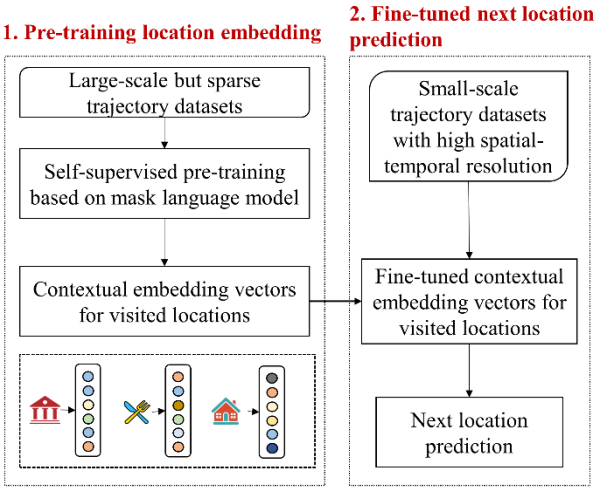

Figure 2. Flowchart of the proposed two-stage next location prediction framework.

Firstly, large-scale sparse location datasets, which are easier to be acquired (i.e., check-in data and anomalous navigation data), are used for pre-training the location embedding model to capture multi-functional properties. Herein, we propose a Context-Aware Spatial Temporal Location Embedding Pre-Training (CASTLE) Model to learn the contextual embedding vectors for visited locations. The same location will have different embedding vectors in different spatial-temporal contexts. After pre-training the CASTLE model, the learned contextual embedding is used for downstream location prediction in small-scale but higher spatiotemporal resolution trajectory datasets. Besides, the parameters of the CASTLE model are fine-tuned to learn the spatial-temporal information in the dataset.

In this paper, a visit $v = (l, t, g)$ indicates that individual visits a location l at time t and the geospatial position of the location l can be denoted as g. Given the trajectory s={(l_1, t_1, g_1), (l_2, t_2, g_2),⋯, (l_n, t_n, g_n)}, the goal of the next location prediction is to predict the output l_{n+1}. And the goal of the pre-training step is to learn a parameterized map function f, which generates the latent contextual embedding vector $V(v_i)$ from a visited record $v_i = (l_i, t_i, g_i)$ and its context C(v_i).

2.2 The Proposed CASTLE Model

Our proposed CASTLE Model (Figure 3) consists of 1) a multi-modal encoding module that inputs the visited location, time, and geospatial position into the model; 2) a bidirectional encoder that learns the embedding of locations by taking other relevant visited locations within the sequence into account. 3) an autoregressive decoder that predicts locations auto-regressively.

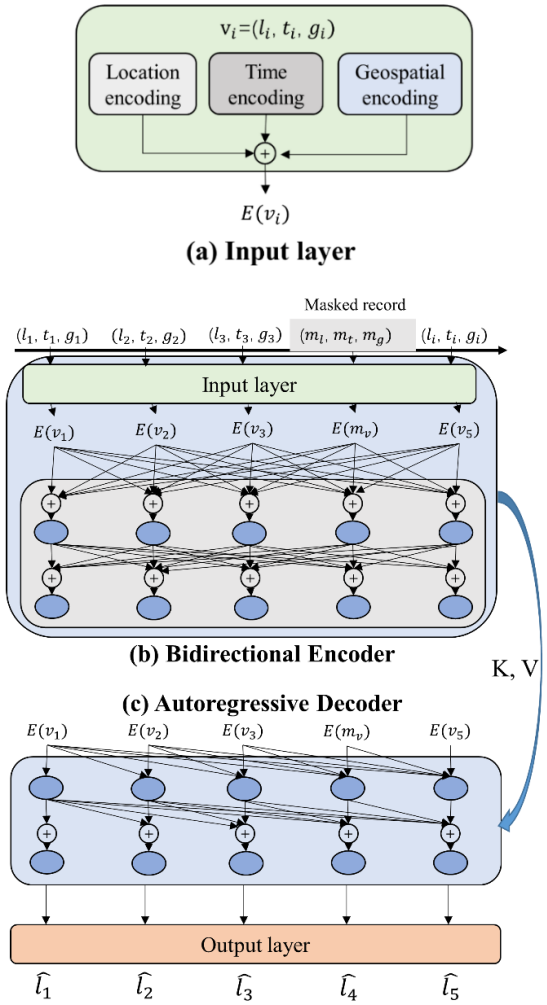

Figure 3. The sketch map of CASTLE. (a) The input layer of CASTLE consists of the location, time, and geospatial position encoding layer. (b) Bidirectional encoder. (c) Autoregressive decoder.

2.2.1 Multi-modal Encoding Module: Given a trajectory $\{v_1 = (l_1, t_1, g_1), v_2 = (l_2, t_2, g_2), \cdots\cdots, v_{n-1} = (l_{n-1}, t_{n-1}, g_{n-1}), v_n = (l_n, t_n, g_n)\}$, the input vector $E(v)$ of visit record v is denoted as:

$$E(v_i) = e(l_i) + e(t_i) + e(g_i), \quad (1)$$

where $e(l_i)$ = embedding vector of location l_i
$e(t_i)$ = embedding vector of time t_i
$e(g_i)$ = embedding vector of geospatial position g_i

(1) location encoding layer: The location encoding is implemented using a fully connected embedding layer, and the embedding layer can be represented as an embedding matrix $Z_l \in \mathbb{R}^{n_l * d_{model}}$, where n_l is the total number of locations and d_{model} is the set dimension of the location embedding vector. A matrix multiplication process $e(l_i) = o(l_i)^T Z_l$ is used to generate the embedding of locations l_i based on the one-hot vector $o(l_i)$.

(2) time embedding layer: Here, the continuous timestamp was mapped into 168 dimensions (7 days * 24 hours per day). Then the time embedding layer can be represented as an embedding matrix $Z_t \in \mathbb{R}^{168 * d_{model}}$.

(3) geospatial position embedding layer: In general, the geospatial position of a visit is usually characterized by latitude and longitude. Nevertheless, this representation method will suffer from the sparsity issue. Thus, we adopt a hierarchical map gridding method to represent the geospatial position (Lian et al., 2020). Because of grid division like quadtrees, each grid was represented as a base-4 number with a certain length (e.g., the length of a quadtree key is 16 at the 16th level of detail). In this way, the spatial distances of different locations can be reflectedd in their quadtree keys.

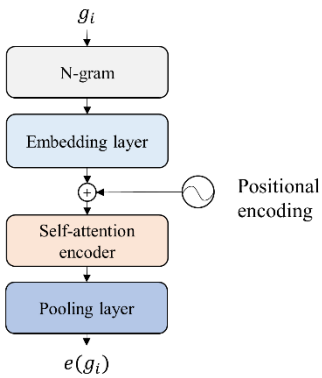

Figure 4. The geospatial encoding layer of CASTLE.

In order to model the spatial positional relationship of visited locations, this study uses the N-gram method and self-attention network to construct a geospatial embedding layer using the tiled quadtree index of trajectory points (Lian et al., 2020). N-gram is a widely used method of segmenting sequences according to a certain length. N-gram consists of a series of substrings obtained by sliding a window of length N by one string at a time. Taking the quadtree index "13101113" as an example, the corresponding trigram sequence when N is 3 is {131, 310, 101, 011, 111, 113}. Since the character set of the quadtree index string only includes {0, 1, 2, 3} four characters, it is not enough for characterizing the whole area. The size of the vocabulary of the embedding layer corresponding to the N-gram is 4^N. In order to obtain the contextual information of sequences in N-grams, the N-gram embedding sequences are represented by a self-attention encoder after adding positional encoding. The self-attention encoder used here is consistent with the encoder in the transformer. Finally, the

embedding of the geospatial position is generated by the average pooling of the n-gram sequence.

2.2.2 Bidirectional Encoder and Autoregressive Decoder: The main body of the CASTLE model adopts the encoder-decoder structure of the transformer (Vaswani et al., 2017). The encoder is adopted to capture sequence context information, and the decoder is used for sequence prediction. The encoder consists of several attention sub-modules with the same layer structure. The bidirectional self-attention in the sub-modules can capture the spatiotemporal context information in the sequence. The output vector of the encoder corresponding to visited records v_i contains the spatiotemporal context information $C(v_i)$. which is represented as $V(v_i)$ in this study. Use the sequence obtained by the decoder as Q, and the representation sequence obtained by the encoder as K and V to perform attention interaction. After multiple self-attention and attention modules, a sequence of latent vectors $\{h_1, h_2, \ldots h_n\}$ with the same dimension as the input vector is finally obtained. Finally, the location is predicted as follows:

$$\hat{l}_i = Softmax(h_i W_f + b), \tag{2}$$

where h_i = the latent output vector of the decoder
$W_f \in \mathbb{R}^{n_l * d}$ and $b \in \mathbb{R}^{n_l}$, both of them are learnable
n_l= the total number of locations
d = the set dimension of the location embedding vector

2.3 Pre-training Objective

The goal of pre-training is to learn a mapping function f to produce a contextual embedding $V(v)$ for a target visit v given its spatial-temporal context $C(v)$.

Inspired by Masked Language Model proposed in BERT, we implement a self-supervised training model. Given a trajectory s, 15% of visited records are randomly chosen as masked visited records and replaced the embedding vectors of masked records with special tokens $[(m_l, m_t, m_g)]$. In the pre-training process, the original visited location of each masked visit record is predicted. And the pre-training objective is expressed as follows:

$$O = \arg\max_{\theta} \sum_{i=1}^{M} p(l_i \mid \hat{l}_i), \tag{3}$$

where θ = all the learnable parameters
M= the number of all the masked visit records
$p(l_i \mid \hat{l}_i)_l$= the probability that location l_i is correctly predicted

2.4 Next Location Prediction Objective

Given the trajectory $\{(l_1, t_1, g_1), (l_2, t_2, g_2), \cdots\cdots, (l_{n-1}, t_{n-1}, g_{n-1}), (l_n, t_n, g_n)\}$, the goal of the next location prediction is to predict l_{n+1} correctly. Thus, the objective of the next location prediction can be represented as:

$$O = \arg\max_{\theta} \sum_{i=1}^{T} p(l_i \mid \hat{l}_i), \tag{4}$$

where θ = the set of all the learnable parameters of the model
T= the number of all the predicted records

The above objectives can be transformed into classification tasks and optimized using the cross-entropy loss function.

3. EXPERIMENTAL RESULT

The experiments were conducted on two real-world spatial-temporal trajectory datasets to verify the effectiveness of the proposed model. Furthermore, the CASTLE model was compared with other models quantitatively.

3.1 Datasets

Two real trajectory datasets were used in the experiments, including a large-scale anomalous navigation dataset and a small-scale mobile phone trajectory dataset in the same city, denoted as TucityPre and TucityLife, respectively. The TucityPre data set was captured from one of the biggest navigation service companies in China and included anonymous mobile phone or vehicle navigation data for about two weeks in May 2021. The TucityLife dataset was collected by some volunteers in the same city in August 2021.

The trajectory-subsequence that a user stays within 100 meters for more than 5 minutes is regarded as visiting the location. We included the trajectories with more than 3 different visit locations in the both two datasets. The numbers of users, locations, and visit records of the two datasets are shown in Table 1.

Dataset	Visit Records	Users	Locations
TucityPre	67082	4033	2354
TucityLife	4266	23	2354

Table 1. Statistics of users, locations, and visit records of the used datasets**.**

3.2 Baseline Models

3.2.1 Baseline Pre-Training Embedding Models: In order to verify the effectiveness of the proposed CASTLE model in pre-training embedding vectors for visits, this paper uses the CTLE model (Lin et al., 2021) as the baseline. The CTLE model uses a BERT-like bidirectional encoder to predict the masked location. The CTLE model interoperates time and location information, which is a state-of-the-art model in pre-training embedding vectors for locations.

3.2.2 Baseline Next Location Prediction Models: To evaluate the usefulness of our framework, we employ some effectivenext location prediction methods:

(1) **GRU** (Cho et al., 2014) (Gate Recurrent Unit): An improved model of the RNN model, we use the GRU-based seq2seq location prediction model as a baseline model.

(2) DeepMove (Feng et al., 2018): a state-of-the-art model consisting of recurrent network and attention layers to capture sequence correlations.

(3) **Pre-trained CASTLE encoder + GRU**: The pre-trained embedding vectors are used as the input to the GRU model.

(4) **CASTLE without pre-training**: Directly train the CASTLE location prediction model on the TucityLife dataset without pre-training.

3.3 Evaluation Metrics

The pre-training embedding method does not have a stable performance evaluation index. Since the downstream task of this study is next location prediction, the trajectory prediction task is used here to evaluate the accuracy of the pre-training model. Specifically, we masked the last visit record in the trajectory sequence and used the pre-training model to predict the last visit location of the trajectory.

Two widely used metrics of next location prediction, including Recall and NDCG (Normalized Discounted Cumulative Gain) (Lian et al., 2020), were adopted in this study. Furthermore, two metrics were both calculated at the cut-off of k = 1 and k = 5.

3.4 Settings

The dimension of location embedding d_{model} was set to 64 for the CASTLE model and other compared methods. We train all the models using the Adam optimizer with a learning rate of 0.001. To avoid over-fitting, we set the dropout ratio to 0.2. In the pre-training process, the mask ratio of the input encoder is set to 15%. Geocoded N-grams use trigrams to represent quadtree indexes and use a two-layer self-attention structure with a single attention head to capture the context of trigrams. Besides, the level of the quadtree key is set as 17 in our experiments. The hidden layer dimension of its feed-forward network is set to 128. The encoder of CASTLE uses a two-layer self-attention structure with four attention heads, the dimension of the feed-forward network hidden layer is set to 128, and the parameter settings of the decoder are consistent with the encoder. The pre-training batch size is set to 64, and the training epoch is set to 1000. In addition, to avoid randomness of the results, a different random seed is used for each epoch training. In pre-training, the first 80% (in time order) of the trajectory visit records for each user are used for training, and the last records are used for testing to prevent data leakage. The parameters in the next location prediction are almost the same as in pre-training. Considering the small size of the TucityLife dataset, the Batch Size is set to 32. For each user's time-ordered trajectory sequence, the first 70% of the trajectory visit records are used as prediction, 20% of the trajecotories are used as validation, and the rest are used as the test set.

3.5 Results

3.5.1 Pre-training Results: After the model was pre-trained on the training set of TucityPre, the trajectory prediction task was chosen to evaluate the performance of the pre-trained CASTLE model. The comparison between the model and the baseline model is shown in Table 2. For the top-1 and top-5 sets of prediction results, the Recall and NDCG of the CASTLE model are both better than the CTLE model. It shows that the CASTLE pre-training method proposed in this study can better capture the spatiotemporal context information by incorporating geospatial position information, which improves the model's accuracy in the trajectory prediction task.

Pre-training model	Recall @1	NDCG @1	Recall @5	NDCG @5
CTLE	0.252	0.252	0.549	0.409
CASTLE	**0.319**	**0.319**	**0.554**	**0.446**

Table 2. Pre-training performance comparison.

3.5.2 Next Location Prediction Results: The location prediction performance of different models on the TucityLife test set is shown in Table 3. By analyzing the next location preddiction results, we couldd conclude the following conclusions:

(1) The performance of the CASTLE model without pre-training is better than that of GRU and DeepMove, indicating that the spatiotemporal context information obtained through attention can play a positive role in location prediction;

(2) Inputting the pre-trained location embedding vectors into the GRU model can also significantly improve the accuracy, indicating that the location embedding vectors pre-trained by the CASTLE model have good transfer performance.

(3) The fine-tuned CASTLE model achieves the best performance and outperforms the model without pre-training by 4.6-7.1%, indicating the effectiveness of our proposed two-stage next location prediction framework,

Model	Recall @1	NDCG @1	Recall @5	NDCG @5
GRU	0.217	0.217	0.439	0.334
DeepMove	0.293	0.293	0.592	0.453
Pre-trained CASTLE encoder + GRU	0.256	0.256	0.532	0.404
CASTLE Without pre-training	0.314	0.314	0.710	0.532
Fine-tuned CASTLE	**0.360**	**0.360**	**0.781**	**0.592**

Table 3. Comparison of next location prediction with different methods.

3.6 Ablation Study

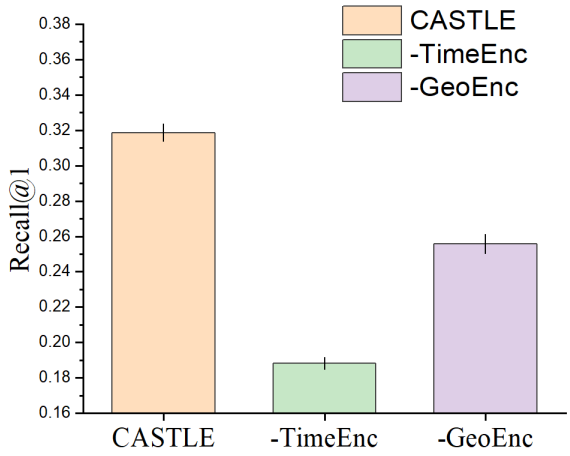

Figure 5. The ablation study results of the CASTLE model.

To further prove the effectiveness of time encoding layer and geospatial encoding layer of the CASTLE model, we design an ablation study, and the compared variants include:

(1) – **TimeEnc**: This model replaces the time encoding layer with the positional encoding layer in Transformers.

(2) – **GeoEnc**: This model just uses the location and time encoding layer.

We compared these two variants with the CASTLE model on the next location prediction task in pre-training. Figure 5 shows that both time geospatial encoding layers benefit the learning of location embedding vectors. The CASTLE model with time and geospatial encoding layers can better capture the spatiotemporal context information of visited locations in the trajectory sequence.

4. CONCLUSIONS

To address the issue that it is challenging to obtain personal trajectories with long time series and high spatiotemporal resolution, we propose a two-stage next-location prediction framework. The first step is pre-training contextual embedding for locations using large-scale trajectory datasets, which are relatively sparse but easier to be acquired. After that, the model is fine-tuned for the next location prediction task in the small-scale but higher spatiotemporal resolution trajectory datasets. Furthermore, a context-aware spatial-temporal location embedding (CASTLE) model is designed for pre-training and next location prediction. The model will generate different embedding vectors for the same location in different spatial-temporal contexts. Specifically, the CASTLE model combines Bidirectional and Auto-Regressive Transformers to predict the next location. Furthermore, we introduce a spatiotemporal aware encoder to reflect the spatial distances between locations and the visit times, which consists of location, time, and geospatial spatial encoding layers. Experiments were conducted on a large-scale anomalous navigation dataset and a small-scale mobile phone trajectory dataset in the same city. The results show that the fine-tuned CASTLE model achieves the best performance and outperforms the model without pre-training by 4.6-7.1%, indicating the effectiveness of our proposed two-stage next location prediction framework. Furthermore, inputting the pre-trained location embedding vectors into other location prediction models can also significantly improve the accuracy, indicating that the location embedding vectors pre-trained by the CASTLE model have good transfer performance. Without massive historical data, our method could still accurately predict the next location in a dense but small-scale trajectory dataset.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under the grant No. 42171327 and the Xinjiang Production and Construction Corps, China under the grant No. 2017DB005.

REFERENCES

Bhargava, P., Phan, T., Zhou, J., Lee, J., 2015. Who, What, When, and Where: Multi-Dimensional Collaborative Recommendations Using Tensor Factorization on Sparse User-Generated Data. In: Proceedings of the 24th International Conference on world wide web,130-140, Florence, Italy. doi.org/10.1145/2736277.2741077.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724-1734, Doha, Qatar. doi.org/10.3115/v1/d14-1179.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171-4186, Minneapolis, Minnesota. doi.org/10.18653/v1/N19-1423.

Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., Jin, D., 2018. DeepMove: Predicting human mobility with attentional recurrent networks. In: Proceedings of the World Wide Web Conference, WWW 2018, 1459-1468, Lyon, France. doi.org/10.1145/3178876.3186058.

Lian, D., Wu, Y., Ge, Y., Xie, X., Chen, E., 2020. Geography-Aware Sequential Location Recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on knowledge discovery & data mining, 2009-2019. doi.org/10.1145/3394486.3403252.

Lian, D., Zheng, K., Ge, Y., Cao, L., Chen, E., Xie, X., 2018. GeoMF: Scalable Location Recommendation via Joint Geographical Modeling and Matrix Factorization. *ACM Trans. Inf. Syst.,* 36 (3), 1-29. doi.org/10.1145/3182166.

Lin, Y., Wan, H., Guo, S., Lin, Y., 2021. Pre-training Context and Time Aware Location Embeddings from Spatial-Temporal Trajectories for User Next Location Prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, 4241-4248. doi.org/10.1609/aaai.v35i5.16548.

Luo, Y., Liu, Q., Liu, Z., 2021. STAN: Spatio-temporal attention network for next location recommendation. In: Proceedings of the World Wide Web Conference, WWW 2021, 2177-2185, Ljubljana, Slovenia. doi.org/10.1145/3442381.3449998.

Monreale, A., Pinelli, F., Trasarti, R., Giannotti, F., 2009. Wherenext: a location predictor on trajectory pattern mining. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 637-646, New York, United States. doi.org/10.1145/1557019.1557091.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, 5999-6009, California, United States. doi.org/10.5555/3295222.3295349.

Wan, H., Li, F., Guo, S., Cao, Z., Lin, Y., 2019. Learning Time-Aware Distributed Representations of Locations from Spatio-Temporal Trajectories. *Database Systems for Advanced Applications*, 268-272. Cham: Springer International Publishing. doi.org/10.1007/978-3-030-18590-9_26.

Wan, H., Lin, Y., Guo, S., Lin, Y., 2021. Pre-training Time-Aware Location Embeddings from Spatial-Temporal Trajectories. *IEEE Tran. Knowl. Data Eng.*, 1-14. doi.org/10.1109/TKDE.2021.3057875.

Wang, Y., Yuan, N.J., Lian, D., Xu, L., Xie, X., Chen, E., Rui, Y., 2015. Regularity and Conformity: Location Prediction Using

Heterogeneous Mobility Data. In: Proceedings of the 21th ACM SIGKDD International Conference on knowledge discovery and data mining, 1275-1284, Sydney, Australia. doi.org/10.1145/2783258.2783350.

Yuan, H., Qian, Y., Yang, R., Ren, M., 2014. Human mobility discovering and movement intention detection with GPS trajectories. *Decis. Support Syst.,* 63, 39-51. doi.org/10.1016/j.dss.2013.09.010.

Zhang, J.-D., Chow, C.-Y., 2014. GeoSoCa: Exploiting Geographical, Social and Categorical Correlations for Point-of-Interest Recommendations. In: Proceedings of the 22nd ACM SIGSPATIAL International Conference on advances in geographic information systems, 443-452, Santiago, Chile. doi.org/10.1145/2766462.2767711.

Zhang, J.-D., Chow, C.Y., Li, Y., 2015. iGeoRec: A Personalized and Efficient Geographical Location Recommendation Framework. *IEEE Trans. Serv. Comput.,* 8 (5), 701-714. doi.org/10.1109/TSC.2014.2328341.