Long-Term User Location Prediction Using Deep Learning and Periodic Pattern Mining

Mun Hou Wong[1], Vincent S. Tseng[1(✉)], Jerry C.C. Tseng[2], Sun-Wei Liu[3], and Cheng-Hung Tsai[3]

[1] National Chiao Tung University, Hsinchu, Taiwan, Republic of China
vtseng@cs.nctu.edu.tw

[2] National Cheng Kung University, Tainan, Taiwan, Republic of China
jerry.cc.tseng@gmail.com

[3] Institute for Information Industry, Taipei, Taiwan, Republic of China
{SunwayLiu, jasontsai}@iii.org.tw

Abstract. In recent years, with the advances in mobile communication and growing popularity of the fourth-generation mobile network along with the enhancement in location positioning techniques, mobile devices have generated extensive spatial trajectory data, which represent the mobility of moving objects. New services are emerged to serve mobile users based on their predicted locations. Most of the existing studies on location prediction were focused on predicting the next location of a user, which is regarded as short-term next location prediction. While more advanced location-based services could be enabled for the users if long-term location prediction could be achieved, the existing methods constrained in next-location prediction are not applicable for long-term prediction scenario. In this paper, we propose a novel prediction framework named LSTM-PPM that utilises deep learning and periodic pattern mining for long-term prediction of user locations. Our framework devises the ideology from natural language model and uses multi-step recursive strategy to perform long-term prediction. Furthermore, the periodic pattern mining technique is utilized to reduce the accumulated loss in the multi-step strategy. Through empirical evaluation on a real-life trajectory dataset, our proposed approach is shown to provide effective performance in long-term location prediction. To the best of our knowledge, this is the first work addressing the research topic on long-term user location prediction.

Keywords: Long-Term prediction · Location prediction · Trajectory pattern mining

1 Introduction

With the fast development of mobile network techniques and the enhancement in location positioning techniques, mobile devices have generated extensive spatial trajectory data, which represent the mobility of moving objects. The ability to predict the future location of a moving object allows for a rich set of innovative services becoming feasible. As a result, predicting the next location of a user given the history spatial trajectory has increasingly gained significance in recent years [1–7].

G. Cong et al. (Eds.): ADMA 2017, LNAI 10604, pp. 582–594, 2017.
https://doi.org/10.1007/978-3-319-69179-4_41

Next location prediction can enhance the effectiveness of advertisement publishing scheme. While integration of location in advertising has demonstrated that it generates considerably higher return than traditional mobile advertising, location based advertisement scheme only utilise the real-time location information of the target customer. This is where the next location prediction can play a role. We believe that if ones can know the likely places where a person will visit, the more valuable context they can get.

Due to the applicability of next location prediction, research on this area has attracted much attention [1–5]. However, most of the existing studies on location prediction were focused on predicting the next location of a user, which is regarded as short-term next location prediction. While more advanced location-based services could be enabled for the users if long-term location prediction could be achieved, the existing methods constrained in next-location prediction are not applicable for long-term prediction scenario. In this paper, we proposed *LSTM-PPM (Long Short Term Memory with Periodic Pattern Mining)*, a new prediction framework based on deep learning and periodic pattern mining that can perform long-term location prediction. We devise the ideology underlying the natural language model and apply it into the location prediction problem. Furthermore, the periodic pattern mining technique is utilised to reduce the accumulated loss in a multi-step strategy we designed. The proposed prediction framework consists of two major steps. Firstly, we pre-process the data and map it into language model. Secondly, we design a hybrid model for predicting user's long-term location at a given time t by means of techniques drawn from deep learning and periodic pattern mining. The contributions of this paper can be summarised as follows:

- We propose *LSTM-PPM*, a new prediction framework that performs long-term location prediction based on deep learning and periodic pattern mining. A multi-step recursive prediction strategy is used in prediction and its accumulated loss is reduced with periodic pattern mining. To the best of our knowledge, this is the first work addressing the research topic on long-term user location prediction.
- We devise the ideology of natural language model and apply it into location prediction. Several data pre-processing techniques were deployed to map the location traces to sentences in language model.
- We evaluate LSTM-PPM by comparing it with Markov predictors on a real-life dataset. We report an overall performance increment over the Markov-based predictor on long-term location prediction.

2 Related Works

Mobility models are a set of well-studied algorithms for next location prediction tasks. Over the course of the last few decades, many mobility model approaches have been developed [1–7]. The mobility models can be grouped into different categories.

Markov Chain (MC) model for next location prediction has been presented in [1, 2, 6]. Markov Chain model is a mobility model that based on probabilistic reasoning to predict the future location of a user using sequences of past observation [8]. The Markov assumption implies that the future predictions are independent of all but the n most recent observations. Markov model can be used to predict a user's next location by computing the transition probability. However, a Markov model with order n can only discover the patterns of length n.

Similarly, [3] employed hidden Markov model to infer user's next location. As described in [8], the Hidden Markov model is a state space model where each observation x_n is attached by a corresponding latent variable z_n. At any time-step n, the states of the latent variable of a hidden Markov model represent a mixture distribution given by the probability $p(x_n|z_n)$, which has generated the observation x_n. A hidden Markov model is used to predict a user's next prediction due to its ability to consider the location characteristics as an unobservable parameter.

[4, 7] proposed a rule-based approach that discovers spatial and temporal patterns from users' trajectory data. After all the frequent patterns have been discovered, they predict users' next location using pattern matching technique.

Recently, some of the studies leverage Recurrent Neural Network (RNN) to model people's mobility [5]. The authors presented that the RNN model archived good results on next location prediction compared to the state-of-the-art methods.

3 The Proposed Framework

Figure 1 shows the proposed framework, namely *LSTM-PPM* (Long Short Term Memory with Periodic Pattern Mining), which consists of three parts, namely the input, process, and output. The input is a set of user's history spatial trajectory and a specified future time t. The process consists of two major parts: the data pre-processing module and the LSTM training. Finally, the output is the predicted location or place semantic of an user at the given time t.

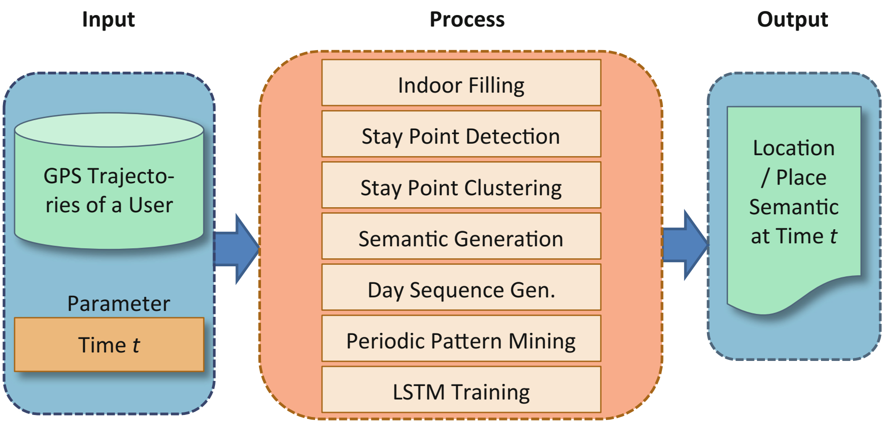

Fig. 1. The proposed framework.

In this section, we first introduce the data pre-processing methods. Next, we will present the whole framework in details.

3.1 Dataset Pre-processing

Indoor Filling. The reliability of GPS data depends on many factors. Most of the time, the GPS track logs collected using a GPS logger exist gaps. This is because GPS signals are often heavily attenuated when users are inside buildings. To overcome this problem, we apply a simple heuristic rule originally from [6]. When the gap between two consecutive GPS records are within a distance of 35 m and the gap duration is greater than 5 min, we consider the user was staying at the same location during that time. Thus, we reconstruct the trajectory by adding as many GPS points equal to the starting point of the gap as the duration in second of the gap.

Stay Point Detection. In a trajectory, not all GPS points are equally important. Some points denote the locations where people have stayed for a while, the other points represent the transition from one place to another. Therefore, we should extract all of his visited places from the raw GPS trajectories, which can be regarded as the locations where the user has stayed. We call these places as stay points. Formally, the data collected by the GPS logger are of the GPS form, which is a sequence of GPS point $P = \{p_1, p_2, \ldots, p_n\}$. Each point $p_i \in P$ contains the longitude, latitude, and time stamp.

A single stay point s can be considered as a group of consecutive GPS points $G = \{p_m, p_{m+1}, \ldots, p_n\}$ such that $distance(p_m, p_i) \leq \varepsilon$ and $time\ difference(p_m, p_n) \geq \tau$, where ε is distance threshold, τ is time threshold and $\forall m < \mathrm{i} < n$. As illustrated in, p1→p2→…→p9 forms a GPS trajectory and a stay point can be constructed by p5→p6→p7→p8. We follow the algorithm proposed in [9] to extract the stay points of a user.

Stay Point Clustering. Occasionally, the detected stay points are too shattered and may not reflect the region where the user stays. Therefore, the detected stay points should be clustered into several groups to represent the possible places where a user has stayed. Thus, we apply the DBSCAN algorithm with these stay points to form stay locations. Each cluster generated by DBSCAN is given a cluster id and the location history of each user is then formulated using these cluster ids.

Place Semantic Inferring. Since a stay location is a cluster of stay points and represented with only geographic coordinates, to retrieve a richer and more meaningful information, we use Google Places API [10] to infer the semantic meaning of each stay location. We search the nearest place to the centroid of each cluster through the Google Places API and associate them with the places type of the nearest place.

Day Sequence Generation. Intuitively, the raw trajectory can be transformed to a sequence of detected stay locations. Hence, we use a predefined time interval length to divide each day into several numbers of equal length time intervals. For example, if the time interval length is set to 30 min, each day can be discretised into 48 windows. Each window is filled with the stay location where it was visited within that time interval (Table 1).

Table 1. An example of day sequence with 30 min time interval.

x^{th} minutes of the day	0^{th}	30^{th}	60^{th}	90^{th}	…	1350^{th}	1380^{th}	1410^{th}
Cluster ID	0	0	0	0	…	1	1	1

It is possible that more than one stay location exists for the same time interval. For such cases, that window is filled with the last visited stay location during that time interval.

Periodic Pattern Mining. Discovering periodic mobility patterns of user can be useful for location prediction task as it unveils repeated activities at regular time intervals. To discover those cyclic patterns, we apply PFPM algorithm [11]. PFPM aims to discover sets of items that periodically appear in transactions. Hence, we treat each location label in the day sequences as a transaction with only one item.

3.2 Design Logic and Implementation Details

[12] demonstrates the power of RNN in language modelling and generating sequences. In language modelling, the input is a sequence of words and the output is the sequence of predicted words. A language model measures how likely a sentence is and predicts the probability of each word given the previous words. An obvious way to generate a sequence is to repeatedly predict what will happen next.

This is my cat. I love my cat.

Fig. 2. Text generation example.

We use Fig. 2 to explain the philosophy behind the text generator. Given a sequence of words "This is my cat." as the input, the text generator tries to generate the next word "I". To generate a word sequence, we append the new generated word to the input and re-run the text generator (Fig. 3).

Sentence	This	is	my	cat.	I	love	my	cat.
Location Trace	1	1	1	2	2	2	3	3

Fig. 3. Mapping from location trace to sentence.

We hypothesise that LSTM could be used in long-term location prediction due to its ability to infer the rich contextual information of the input locations. It can learn the hidden contextual meaning of the location labels in the location traces. We model the location traces of each user as sentences and each location label as a word. The location labels correspond to words in the language model. The entire location trace can be mapped into sentences, where the prediction of the next location is the same as the prediction of next word.

Figure 4 depicts our proposed LSTM architecture. In our LSTM model, each trajectory is reformed as a day sequence and represented as a sequence of tuple $(\mathbf{x}_\tau, \mathbf{h}_\tau, \mathbf{d}_\tau)$, where $\mathbf{x}$ is the centroid ID of the stay location visited at time τ, while $\mathbf{h}$ and $\mathbf{d}$ are the associated beginning time of the interval and day of week. Its task is to predict the future location of a user during the next time interval.

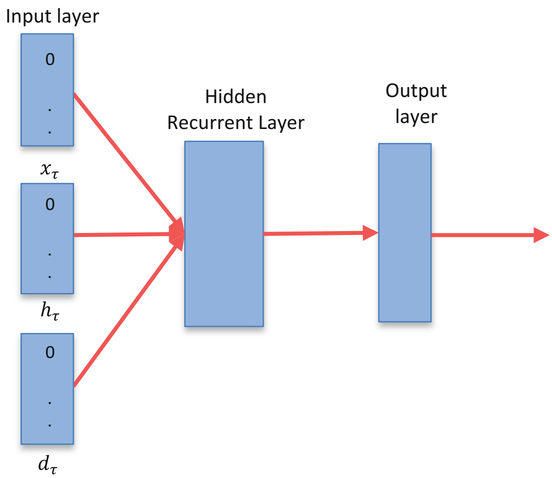

Fig. 4. Proposed LSTM architecture.

Our proposed LSTM model consists of three layers: input layer, hidden recurrent layer and output layer. The input layer consists of three vectors. The first one is $\mathbf{x}_\tau \in R^N$ which represents the centroid ID of the stay location at time τ, where N is the number of all stay location. The second vector represents the beginning time of the interval at time t, denoted as $\mathbf{h}_\tau \in R^M$, where M is the number of time intervals in day sequence. Finally, the third vector, denoted as $d_\tau \in R^L$ represents the associated day of week of time τ. L is the number of days in a week. Each vector in the input layer is encoded using one-hot encoding.

We use LSTM recurrent unit in hidden layer to learn and capture the latent information of user mobility behaviour. As mentioned before, we apply LSTM in modelling the sequence and use its ability to connect previous locations to present location. At last, the output layer $\mathbf{y}_\tau \in R^N$ is a categorical classification layer, produces the probability of all the locations and it has the same dimensionality as the input vector $\mathbf{x}_\tau$.

Since the day sequence we put into the LSTM model is a sequence with fixed time interval and our model always outputs the next location where the user may visit for the next equal time interval, to predict a location at a given time t, we use a recursive multi-step prediction strategy. The recursive strategy uses the prediction for the prior time step as an input for making a prediction on the following time step. Specifically, we first calculate the number of time steps the model needs to predict, denoted as K. Then, for each prediction, the previous output is treated as the new input for the model, until K predictions have been done. The last output is considered as the prediction result for the given time t.

However, since the predictions are used as observations, the recursive strategy accumulates prediction errors and performance can quickly degrade as the time-step increases. We argue that human mobility patterns are regular. Hence, we combine the periodic pattern mining and LSTM model to overcome the shortfall. For instance, given that the current time is 8:00 A.M., if we are asked to predict where the user might be at 4:00 P.M., our LSTM may not give an accurate prediction since the number of time-steps is large for recursive strategy. However, if we could know that the user will visit a restaurant A at 3:00 P.M. due to his regularity beforehand, the prediction error can be lower since the number of necessary prediction time-steps from 3:00 P.M. to 4:00 P.M. is relatively small. This is where the periodic pattern mining plays its role.

Figure 5 depicts our prediction scheme. We first get all location and their period through periodic pattern mining in Line 1. For each pattern we get, we find the most recent time when that location got visited in Line 4. Line 6–7 finds the furthest time that location will be visited again through our assumption on mobility regularity. Line 8–16 predicts the location through the recursive multi-step forecast strategy. Finally, our algorithm returns the most possible location predicted through majority voting.

```
Algorithm MakePrediction(L, D, ct, t)
Input:   An LSTM model L, all user's day sequences D, current time ct,
         expected time t
Output:  Possible location at t
1.    patterns = PPM(D)  // get periodic patterns
2.    possible_locations = []
3.    for pattern in patterns:
4.        last_occur_time = get_most_recent_occur(D, ct, pattern.location)
5.        possible_occur_time = last_occur_time
6.        while possible_occur_time < t:
7.            possible_occur_time += pattern.period
8.        require_time_step = calculate_time_step(possible_occur_time, t)
9.        h = possible_occur_time.h
10.       d = possible_occur_time.d
11.       while k < require_time_step:
12.           predict_location = L.predict(x, h, d)
13.           x = predict_location
14.           h = update_h(h)
15.           d = update_day_of_week(d)
16.           k += 1
17.       possible_locations.append(x)
18.   return majority_voting(possible_locations)
```

Fig. 5. LSTM-PPM prediction scheme.

4 Experiment Evaluation

4.1 Data Description

We evaluate our framework using Geolife dataset [13], a real-life GPS trajectories dataset. This GPS trajectory dataset was collected in Geolife project initiated by Microsoft Research Asia. It collected trajectories of 182 users in a period of over five years, beginning from April 2007 to August 2012. Each GPS trajectory is represented by a sequence of time-stamped points, containing absolute latitude, longitude, and latitude in fine granularity. These trajectories were logged in a dense representation by different GPS loggers and GPS phones.

4.2 Experiment Settings

Geolife dataset is a dataset built for transportation prediction task. Consequently, not all trajectories contain stay locations. Hence, we selected users under two considerations: the period a trajectory spans and the number of days which all trajectories of a single user cover. In particular, for each user, we only consider the daily trajectories that are recorded for more than 3 h. Under this premise, we consider only the users with more than 90 days of data. The resulting number of users is 11. In addition, since visited locations, mobility, and moving trajectories are different among users, we built one prediction model for each user.

We use Hyperopt [14] to find the optimal hyper-parameters of the recurrent model. Hyperopt is a Python library that use both random search and Tree of Parzen Estimators (TPE) algorithms to search for a set of optimal hyper-parameters in a search space with real-valued, discrete, and conditional dimensions. Finally, we use the first 60 days of data as training dataset and evaluate the model performance with the remaining 30 days of data.

4.3 Comparison Targets and Metrics

We compare our proposed framework with Markov Chain model, which was introduced in [1] and built based on the contextual co-occurrences between sequences of locations.

To assess the performance of our proposed framework, we use the Recall score as an evaluation metric in all experiments. The Recall@K is defined as the ratio between the number of correct predictions over the total number of prediction. We first ranked a list of all potential locations arranged in descending order according to their probabilities. Then the Recall score is calculated as the percentage of times in which the real visited location was found in the top K most likely location in the ranked list:

$$Recall@K(\%) = \frac{Number\ of\ instances\ predicted\ correctly\ at\ TOP\ K}{Number\ of\ instances}$$

4.4 Performance Evaluation

We evaluate the performance of our proposed method by performing, long-term location prediction, in which we predict the user's location for the forthcoming 100 to 500 h. Meanwhile, each group of experiments predicts both the location label and location semantic.

4.4.1 Long-Term Prediction Experiments

Figures 6, 7, 8, 9, 10 and 11 report the performance of the LSTM-PPM on long-term prediction. For long-term prediction, the LSTM-PPM prediction scheme significantly reveals the advantages of periodicity of human mobility. Our prediction scheme generally performs better than the LSTM-only and Markov Chain prediction scheme. On the other hands, the LSTM-only prediction scheme performs better than the Markov Chain prediction scheme for every prediction hour in long-term prediction experiments. This is due to the incapability of Markov Chain to handle the unseen combination of location as input.

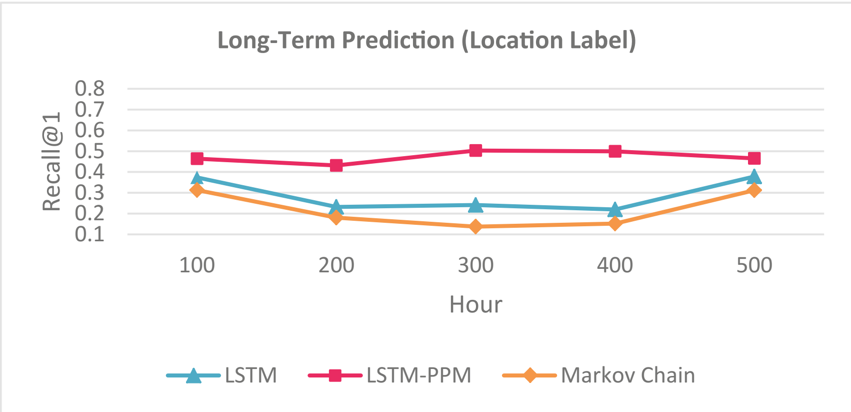

Fig. 6. Long-term prediction (label) Recall@1.

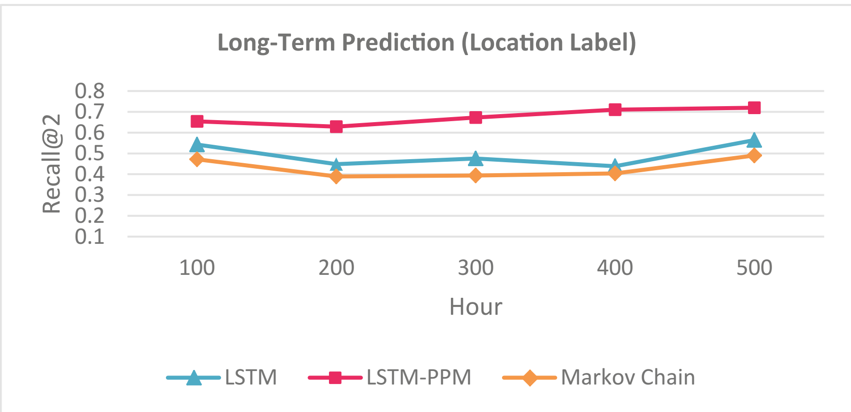

Fig. 7. Long-term prediction (label) Recall@2.

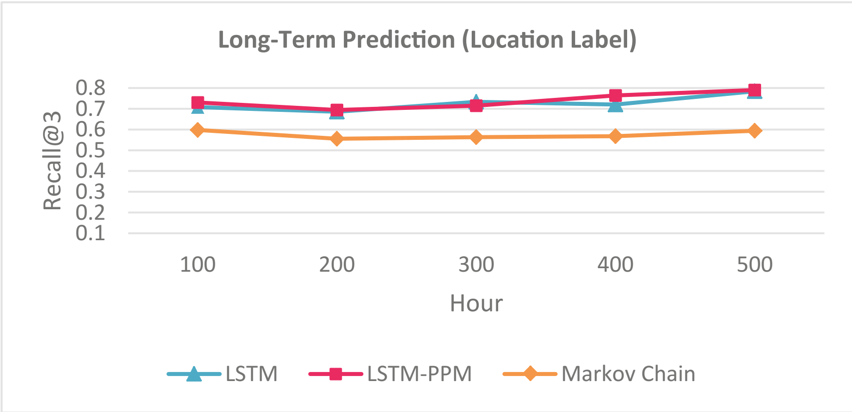

Fig. 8. Long-term prediction (label) Recall@3.

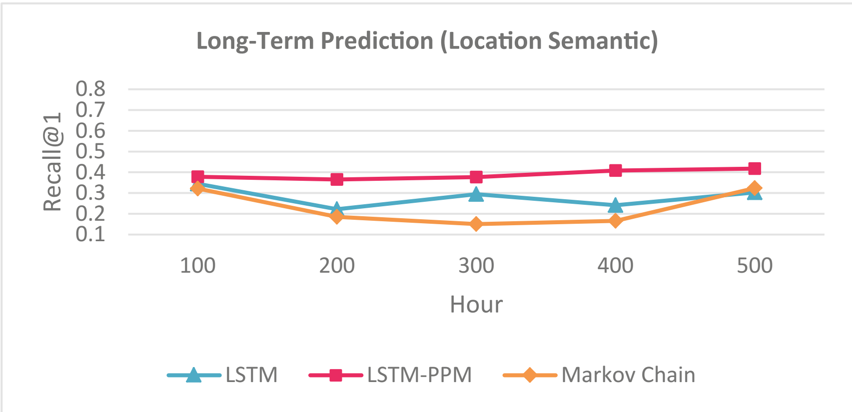

Fig. 9. Long-term prediction (semantic) Recall@1.

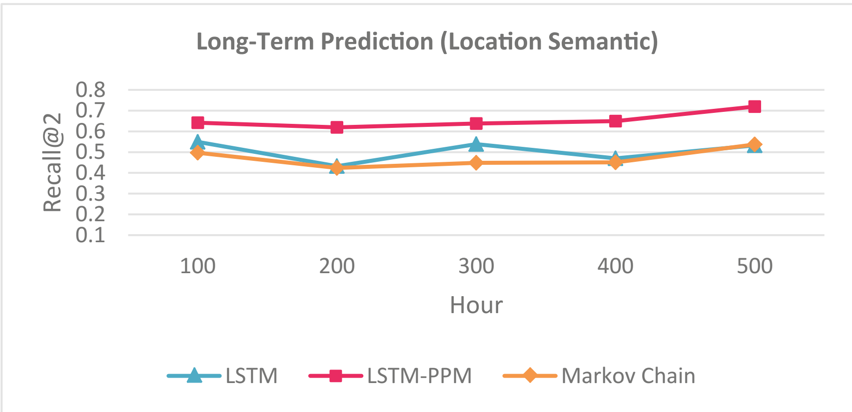

Fig. 10. Long-term prediction (semantic) Recall@2.

4.5 Discussions

Intuitively, the recall score for predicting the semantic should be better than predicting the exact location label. For instance, three location clusters are given three different labels, but they might share the same semantic meaning. However, in either prediction scheme, the recall score for predicting the location label is better than predicting the location semantic, in contrast. We infer that this is due to the inaccurate semantic inferring.

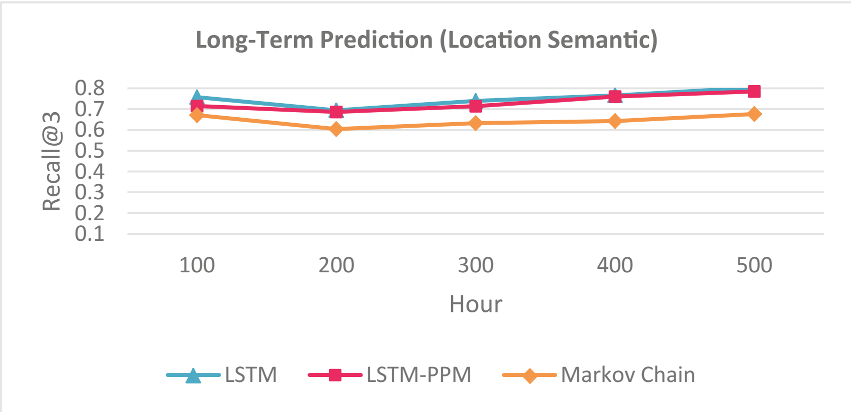

Fig. 11. Long-term prediction (semantic) Recall@3.

(a) (b) (c)

Fig. 12. Same GPS point on different map providers.

Figure 12 shows a same GPS point on three different map providers: (a) Google Maps, (b) OpenStreet Map, and (c) Baidu Map. We plot the GPS coordinate, (39.975011, 116.333481) on these three maps and different semantic meaning of that point could be inferred from the information they give. We believe that this ambiguity is the reason why the semantic prediction performs worse.

5 Conclusion

In this paper, we addressed the topic of long-term user location prediction and investigate the applicability of Long Short-Term Memory Network built upon the analogy between human mobility and natural language models. We have proposed a novel prediction framework with several core ideas: First, to utilise the LSTM network, we apply several data pre-processing techniques to extract the significant locations and transform the trajectories into day sequences to match the ideology of language model. Second, we devised a multi-step recursive prediction strategy based on the idea of text generation to perform prediction. Third, to conquer the loss issue in multi-step recursive prediction strategy, we utilise the periodic pattern mining technique to reduce the number of required time steps for prediction and thus reduce the accumulated loss.

Through experimental evaluation on a real dataset, our proposed framework was shown to deliver good performance compared to other representative methods. To sum up, this paper achieves several contributions: First, we devise a data pre-processing method to make the trajectories fit into the architecture of LSTM. Second, we proposed a multi-step recursive strategy that enables our prediction scheme for long-term prediction. Finally, we combine the periodic pattern mining technique to reduce the accumulated loss in multi-step recursive strategy and make the whole approach appealing in terms of the accuracy. To the best of our knowledge, this is the first work addressing the research topic of long-term user location prediction with providing an appealing approach. We believe that many innovative location-based applications can be enabled through the proposed approach.

Acknowledgement. This study is partially supported under the "System-of-systems Driven Emerging Service Business Development Project (2/4)" of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China.

References

1. Asahara, A., Maruyama, K., Sato, A., Seto, K.: Pedestrian-movement prediction based on mixed Markov-Chain model. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 25–33. ACM (2011)
2. Gambs, S., Killijian, M.-O., del Prado Cortez, M.N.: Next place prediction using mobility Markov Chains. In: Proceedings of the First Workshop on Measurement, Privacy, and Mobility, p. 3. ACM (2012)
3. Mathew, W., Raposo, R., Martins, B.: Predicting future locations with hidden Markov models. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pp. 911–918. ACM (2012)
4. Lee, S., Lim, J., Park, J., Kim, K.: Next place prediction based on spatiotemporal pattern mining of mobile device logs. Sensors **16**, 145 (2016)
5. Liu, Q., Wu, S., Wang, L., Tan, T.: Predicting the next location: a recurrent model with spatial and temporal contexts. In: AAAI, pp. 194–200 (2016)
6. Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. Pers. Ubiquit. Comput. **7**, 275–286 (2003)

7. Ozer, M., Keles, I., Toroslu, H., Karagoz, P., Davulcu, H.: Predicting the location and time of mobile phone users by using sequential pattern mining techniques. Comput. J. **59**, 908–922 (2016)
8. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
9. Ye, Y., Zheng, Y., Chen, Y., Feng, J., Xie, X.: Mining individual life pattern based on location history. In: International Conference on Mobile Data Management: Systems, Services and Middleware (MDM), pp. 1–10. IEEE (2009)
10. Google. https://developers.google.com/places/?hl=en-gb
11. Fournier-Viger, P., Lin, C.-W., Duong, Q.-H., Dam, T.-L., Ševčík, L., Uhrin, D., Voznak, M.: PFPM: discovering periodic frequent patterns with novel periodicity measures. In: Proceedings of the 2nd Czech-China Scientific Conference 2016. InTech (2017)
12. Graves, A.: Generating Sequences with Recurrent Neural Networks. Arxiv Preprint arXiv:1308.0850 (2013)
13. Zheng, Y., Xie, X., Ma, W.-Y.: Geolife: a collaborative social networking service among user, location and trajectory. IEEE Data(base) Eng. Bull. **33**, 32–39 (2010)
14. Bergstra, J., Yamins, D., Cox, D.: Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: International Conference on Machine Learning (ICML), pp. 115–123 (2013)