# CLoSe: Contextualized Location Sequence Recommender

Ramesh Baral, S. S. Iyengar, Tao Li
School of Computing and Information Sciences
Florida International University
Miami, FL
Emails:(rbara012,iyengar,taoli)@cs.fiu.edu

N. Balakrishnan
Indian Institute of Science
Bangalore, India
Email:balki@serc.iisc.in

## ABSTRACT

The location-based social networks (LBSN) (e.g., Facebook, etc.) have been explored in the past decade for Point-of-Interest (POI) recommendation. Many of the existing systems focus on recommending a single location or a list which might not be contextually coherent. In this paper, we propose a model termed **CLoSe** (**C**ontextualized **Lo**cation **Se**quence Recommender) that generates contextually coherent POI sequences relevant to user preferences. The POI sequence recommenders are helpful in many day-to-day activities, for e.g., itinerary planning, etc. To the best of our knowledge, this paper is the first to formulate contextual POI sequence recommendation by exploiting Recurrent Neural Network (RNN). We incorporate check-in contexts to the hidden layer and global context to the hidden and output layers of RNN. We also demonstrate the efficiency of extended Long-short term memory (LSTM) in sequence generation. The main contributions of this paper are: (i) it exploits multi-context, personalized user preferences to formulate contextual POI sequence generation, (ii) it presents contextual extensions of RNN and LSTM that incorporate different contexts applicable to a POI and POI sequence, and (iii) it demonstrates significant performance gain of proposed model on pair-F1 and NDCG metrics when evaluated with two real-world datasets.

## KEYWORDS

Information Retrieval; POI Recommendation; Social Networks

**ACM Reference Format:**
Ramesh Baral, S. S. Iyengar, Tao Li and N. Balakrishnan. 2018. CLoSe: **C**ontextualized **Lo**cation **Se**quence Recommender. In *Proceedings of Twelfth ACM Conference on Recommender Systems (RecSys '18).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3240323.3240410

## 1 INTRODUCTION

The POI recommenders exploit check-in information and some implicit and explicit contexts to infer preferable POIs. Most of the existing systems focus on generating some POIs [1–6] that match user preferences. Often, day-to-day activities (e.g., itinerary planning) require semantically coherent list of items that satisfy user preferences. Such a coherent list can be formulated as a sequence modeling problem where the sequence of items need to adhere to users' preferences and constraints applicable to consecutive items and also to the whole sequence (e.g., distance between consecutive POIs, total travel time, etc).

Unlike the general recommenders, the POI recommender involves various contexts (e.g., social, temporal, etc.) [1, 2], and is more challenging. Besides the contextual challenges, the POI sequence generation has following additional challenges: (i) computing all permutations of the POIs is NP-hard problem, (ii) getting a preferred list from simple lists may not be optimal because the preferred list need to satisfy multiple constraints, e.g., the POIs should be contextually coherent, need to satisfy spatial (near/far places), temporal (relevant to time of a day, e.g., bars can be relevant at evening or night), social (trip might be a family, friend focused), time budget, start/end POIs, etc., and (iii) the contexts can be user dependent and can vary dynamically in real-time. To the best of our knowledge, there are some POI sequence recommenders most of which either use simple frequency-based user interest, average visit duration or use only a few constraints and ignore many other useful constraints. We attempt to fill the gap by defining and incorporating multiple constraints on neural network-based POI sequence model.

Inspired by the wider popularity of RNN and its variants in text and image domain [7–9], we attempt to cope with the challenges of contextual POI sequence modeling by incorporating the local contexts (context valid for subsequence) and global contexts (context valid for whole sequence). The local contexts (known as context in rest of this paper) are incorporated into the recurrent module and the global contexts (known as feature of sequence in rest of the paper) are fed to all the layers in the network. We also introduce contextual LSTM for personalized POI sequence modeling. To the best of our knowledge, the proposed model is the first one to address multi-context POI sequence modeling using extended RNN and LSTM. The major contributions of this paper are: (i) it incorporates major personalized user constraints (e.g., temporal, spatial etc.) in POI sequence modeling, (ii) it formulates multi-context POI sequence modeling with RNN and LSTM, and (iii) it demonstrates the efficiency of proposed models on two real-world datasets.

## 2 RELATED RESEARCH

The collaborative filtering [10, 11], apriori principle [11, 12], content-based [13], matrix factorization [14], topic-modeling [15, 16], heuristic-based modeling [11, 17], tree traversal [10], and hybrid [18] techniques are heavily used by existing systems. The greedy approach-based model [19] matched user profiles (consisting of preference on categories, keywords, etc.) to relevant places. The ranking model [20] personalized travel sequences in different seasons by merging textual data and viewpoint information extracted from images but


*RecSys '18, October 2–7, 2018, Vancouver, BC, Canada*

ACM ISBN 978-1-4503-5901-6/18/10…$15.00
https://doi.org/10.1145/3240323.3240410

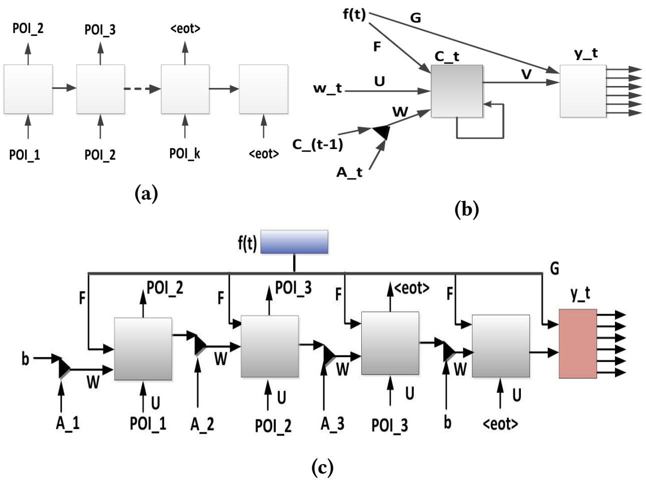



**Figure 1: (a) RNN Sequence modeling, (b) Contextual sequence modeling, (c) Unraveled view of (b)**

ignored social and temporal preferences, and temporal popularity aspects. Wang et al. [21] handled crowd constraints (i.e., peak hour of POIs) by extending the Ant Colony Optimization algorithm. A probabilistic model [22] used Rank-SVM to rank the items and used Markov model to predict the transition between POIs. Lim et al. [23] exploited geo-tagged images and contexts, such as visit duration, users' preferences, and start/end POIs to define time-based user preferences. However, they did not address the categorical, temporal, and social constraints on the POI sequence. The Recurrent Recommender Networks [24] predicted future behavioral trajectories on user-movie preference dynamics by exploiting LSTM [25]. Most of the existing studies have exploited few contexts and have focused on personalized POI visit durations. Unlike others, our study exploits multiple contexts (temporal, spatial, categorical, and social) and formulates the multi-context POI sequence modeling with RNN and LSTM.

## 3 METHODOLOGY

The set of scenarios having (in)direct influence on next POI selection is called context and can be represented as a high-dimensional vector. Given a set of contexts C = $\{\vec{c_i}\}$, our goal is to predict a sequence of POIs relevant to context and user preferences. We define a user's travel history as chronological check-ins $H_u = (V_1, V_2, ..., V_n)$, where $V_i = (l_i, a_i, d_i)$ is a triplet with POI ($l_i$), arrival time ($a_i$) to $l_i$, and the departure time ($d_i$) from $l_i$. We select the time interval of 8 hours [19] to split the travel sequences.

The average stay time (ST) of a POI $i$ is defined as:

$$ST(i) = \frac{1}{\mid U \mid} \sum_{\substack{u \in U \\ i \in L_u}} \frac{1}{\mid V_{u,i} \mid} \sum_{l \in V_{u,i}} \Big( a_{l+1} - TT(l, l+1) - a_l \Big), \quad (1)$$

where U is the set of all users, $L_u$ is the set of locations visited by user $u$, $\mid V_{u,i} \mid$ is the number of check-ins of user $u$ to location $i$, and TT(a,b) is the travel time between POI $a$ and POI $b$. We use a log normal distribution to compute the travel time between consecutive check-ins. $ST'(i)$ denotes 0-1 normalized stay time. The user interest to a place is defined in terms of aggregate stay time (AST) to that place and is derived from its stay time, visit frequency, and the stay time to all places of same category.

$$AST(u,i) = (1-\alpha) * \Big( \sum_{i \in L_u} \frac{ST'(i)}{\mid V'_{u,i} \mid} \Big) + \alpha * \sum_{\substack{l \in L_u \\ l.cat = i.cat}} \frac{ST'(l)}{\mid V'_{u,l} \mid}, \quad (2)$$

where $l.cat$ is the category of location $l$, $V'_{u,l}$ is the 0-1 normalized form of $V_{u,l}$, and $0 \leq \alpha \leq 1$ is a constant factor estimated using the fraction of check-ins that are of same category as $i$. The average stay time with social impact is defined as:

$$AST(u,i) = (1 - \psi_1) * AST(u,i) + \psi_* \sum_{k \in u_f} AST(k,i), \quad (3)$$

where $u_f$ is the set of friends of $u$, $0 \leq \psi_1 \leq 1$ is estimated using the fraction of check-ins of user $u$ that are common to her friends. The average stay time by a user $u$ to a category *'cat'* is defined as:

$$AST(u,cat)_{cat} = (1-\gamma_1) * \Big( \sum_{\substack{i \in L_u \\ i.cat = cat}} \frac{ST'(i)}{\mid V'_{u,i} \mid} \Big) + \gamma_1 * \sum_{\substack{j \in u_f, \\ k \in L_j, k.cat = cat}} \frac{ST'(k)}{\mid V'_{j,k} \mid}, \quad (4)$$

where $\gamma_1$ is a tuning factor estimated using the fraction of check-ins of user $u$ that are common to her friends and have category *'cat'*. The preference score (PS) of user $u$ to a POI $l$ at time $t$ is defined as:

$$PS(u,l,t) = \beta * \Big( (1-\theta) * \sum_{l \in L} \frac{\mid V'_{u,l,t} \mid}{\mid V'_{u,l} \mid} + \theta * \sum_{\substack{l' \in L \\ l'.cat = l.cat}} \frac{\mid V'_{u,l',t} \mid}{\mid V'_{u,l'} \mid} \Big) + (1-\beta) * AST(u,l), \quad (5)$$

where $V_{u,l,t}$ is the number of visits from user $u$ to location $l$ at time $t$ and $V'_{u,l,t}$ is its normalized form, $0 \leq \theta \leq 1$ is estimated as in Eqn. 2, and $0 \leq \beta \leq 1$ is a tuning factor estimated using TF-IDF (term frequency inverse document frequency) on stay time and temporal categorical preferences. The generalized preference score $PS(l,t)$ uses visit frequencies and stay time of all users to $l$. The constraints (e.g., distance, cost, etc.) penalized preference of a place is defined as: $P(u,l,t) = PS(u,l,t) * \frac{1}{\sum_{i=1}^{m} Constraint_i(l,p)}$, where $Constraint_i(l,p)$ is a normalized numeric measure of $i^{th}$ constraint between the current location $p$ and the target location $l$. The above mentioned preference metrics are used as features and attributes (defined later) when we train our prediction model.

**Preliminaries:** RNNs are ideal for sequence modeling because they can efficiently model the generative process of sequential information, summarize the information in hidden states, and generate new sequences by using probability distribution specified by the hidden states. The hidden state $h_t$ from the input sequence $(x_1, x_2, ...., x_t)$ is recursively updated using non-linear transformation function (e.g., tanh). The probability of a sequence $\overrightarrow{x} = x_1, x_2, ...., x_N$ is defined as: $p(\overrightarrow{x}) = \prod_{t=1}^{N} p(x_t \mid h_{t-1})$. The vanishing gradient problem (i.e., a problem with gradient-based learning methods where the gradient decreases exponentially with the value of n (number of layers) while the front layers train very slowly, see [26] for detail) of vanilla RNN are addressed by architectural extensions (e.g., LSTM [25]) and context enriched models [7]. The regular RNNs can use probability distribution specified by the hidden state to predict the next POI. The hidden state summarizes the information of the sequence observed so far (e.g., $l_1, l_2, ...., l_{t-1}$). A hidden layer depends on the current input $x_t$ and the input at previous step $h_{t-1}$, i.e. $h_t = f(W_{xh} x_t + W_{hh} h_{t-1})$, where $W_{xh}$ and $W_{hh}$

are weights for current input, and for the value from previous hidden layer respectively. The probability of a POI sequence can be then defined using the chain rule: $p(\overrightarrow{l}) = \prod_{t=1}^{T} p(l_t \mid h_{t-1})$, where T is the number of timesteps to be considered. The negative log likelihood is used as the sequence loss to train the network: $\mathcal{L}(x) = -\sum_{t=1}^{T} log(p(l_t \mid h_{t-1}))$. For the defined network, the partial derivative of the loss is computed using the backpropagation through time [27] and the network is trained using gradient descent. Such a sequence modeling is illustrated in Figure 1a. This network predicts the next POI by learning from the previously fed POI sequences only, and ignores other information, such as POI attributes, current context, and other relevant features.

**Contextual POI Sequence Modeling using RNN (CLoSe-RNN):** This model is illustrated in Figure 1b and Figure 1c. Both the input and output vectors of our models have dimension of the number of POIs. The input vector $w(t)$ is the check-in at time t (which is one hot encoding), and the output layer produces a probability distribution over the POIs, given the context, feature and the previous check-in. Inspired from [7], we extend the vanilla RNN to incorporate the context and feature inputs (i.e. f(t)). The context and attributes of current input, the current input itself, and the features representing the whole sequence are fed to the network. The contexts are propagated to the hidden layer and features are propagated to all the layers. In addition to context and feature, the attribute vector which represents any information that is relevant to the current item (e.g., location preference score, average stay time, etc.) is also incorporated to the network.

The extension helps us to propagate relevant contexts and features which the network uses at each and every iteration (see Figure 1b). The hidden layer and the output layer are updated as:

$$c(t) = \mathcal{H}(Uw(t) + W(c(t-1) \odot A(t)) + Ff(t)),$$
$$\hat{y}_t = Vc(t) + Gf(t), y(t) = g(\hat{y}_t), \quad (6)$$

where $\mathcal{H}(.)$ is a non-linear function (e.g., tanh), A(t) is the context attribute vector for time $t$, f(t) is the sequence feature vector, and $g(a_i) = \frac{\exp(a_i)}{\sum_j \exp(a_j)}$ is the softmax function. After we train the network (e.g., using stochastic gradient descent), the output vector y(t) gives the probability of occurrence of POIs at time $t$, given the previous POI, context, and the feature vector. The sequence feature vector contains information relevant to the whole sequence. $f(t) = \langle cat_{start}, cat_{end}, loc_{start}, loc_{end}, loc_{dist}, time_{start}, time_{end} \rangle$ is the feature vector, where the terms *cat, loc, time* denote the category, location and time and the subscripts $_{start}$, $_{end}$, and $_{dist}$ denote start, end, and distance respectively for POI sequence (e.g., $cat_{start}$ is starting category). The attribute vector: $A(u, l, t) = \langle ST'(l), AST(l), AST(u, t)_{cat}, PS(l), l.cat, l_T, l_{dist} \rangle$, contains information relevant to current item context, where $l_T = \langle l_1, l_2, .., l_T \rangle$ is hourly popularity of location $l$ and $l_{dist}$ is the distance of location $l$ from the previous location in the sequence (or distance from place with most frequent check-in (e.g., home) of user $u$). The feature vector and attribute vector can be used to incorporate additional features and attributes if required. The network is trained to learn the weight matrices (U, V, W, F, and G), and to maximize the likelihood of the training data [9]. The probability of a POI sequence $l$ for the network is then defined as: $p(l) = \prod_{t=1}^{T} p(l_{t+1} \mid y_t)$.

**Contextual POI Sequence modeling using LSTM (CLoSe-LSTM):** The core strength of LSTM is the memory state and multiple gating functions that control the write, read, and removal (forget) of the information from memory state. The information is propagated by applying these gates to the data from input and previous memory states. We incorporate explicit context to each LSTM cell because each cell models a subsequence and each subsequence can have potentially unique context. This explicit context to each LSTM cell and the feature of sequence to all LSTM cells help us propagate the relevant context for the sequence modeling. Due to space constraint, we only provide the update equations of our extended contextual LSTM:

$$i_t = \sigma(W_i x_t + W_{hi}(h_{t-1} \odot A_t) + W_{ci} c_{t-1} + b_i + W_f F),$$
$$f_t = \sigma(W_f x_t + W_{hf}(h_{t-1} \odot A_t) + W_{cf} c_{t-1} + b_f + W_f F),$$
$$z_t = tanh(W_z x_t + W_{hc}(h_{t-1} \odot A_t) + b_z + W_f F),$$
$$c_t = f_t \odot c_{t-1} + i_t \odot z_t,$$
$$o_t = \sigma(W_o x_t + W_{ho}(h_{t-1} \odot A_t) + W_{co} c_{t-1} + b_o + W_f F),$$
$$h_t = o_t \odot tanh(c_t), \quad (7)$$

where $c_t$ is the memory state, $z_t$ is the module that transforms information from input space $x_t$ to the memory space, and $h_t$ is the information read from the memory state. The input gate $i_t$ controls information from input $z_t$ to memory state, the forget gate $f_t$ controls information in the memory state to be forgotten, and the output gate $o_t$ controls information read from the memory state. The memory state $c_t$ is updated through a linear combination of input filtered by the input gate and the previous memory state filtered by the forget gate. The term $W_f$ is feature weight matrix, $F$ is feature vector, $A_t$ is attribute vector as in the contextual RNN model, and $\odot$ is element-wise product operator. The relevant weight matrices $W$ and biases $b$ are subscripted accordingly.

**Sequence generation:** We train the network using a sequence data one step at a time. The sampled output from the network is then fed as the input for the next level. If we have K different POIs and POI k is checked-in at time $t$, then the input $x_t$ is one-hot encoded vector with only the $k^{th}$ entry set to 1. The output of the network is a multinomial distribution that is parameterized using a softmax function and is defined as: $p(x_{t+1} = k \mid y_t) = y_t^k = \frac{\exp(\hat{y}_t^k)}{\sum_{k'=1}^{K} \exp(\hat{y}_t^{k'})}$.

From the generated sequences, the top-k scorers (sum of preference scores (PS) of all places in a sequence) are recommended to user.

## 4 EVALUATION

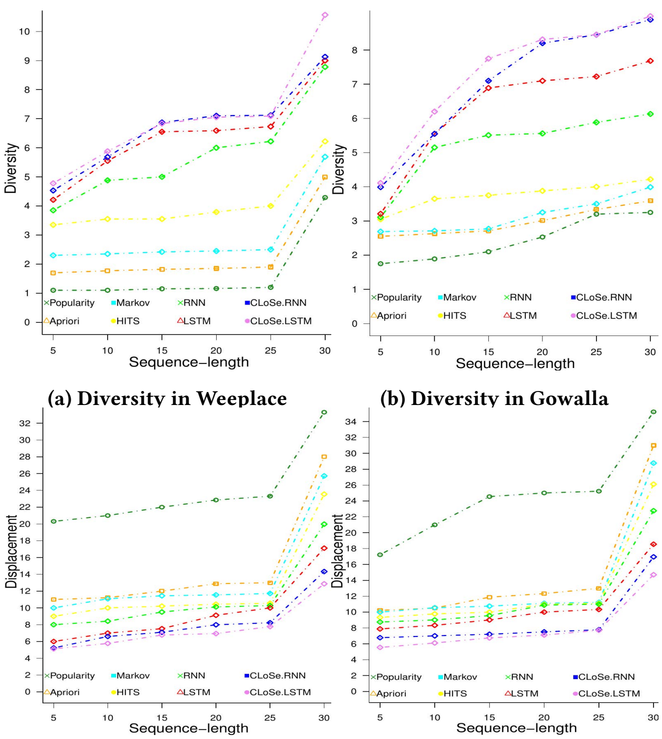
We use Weeplaces[1] and Gowalla LBSN dataset [28]. The former dataset has 7,658,368 check-ins of 15,799 users on 971,309 different locations and the latter one has 36,001,959 check-ins of 319,063 users on 2,844,076 locations. The datasets are well defined and relevant. The *diversity* of a sequence uses the categorical similarity (i.e. Similarity =1, only if two POIs have same category, else it is 0) and gives the number of categories of POIs in a sequence: Diversity $(c_1, c_2, ..., c_n) = (\sum_{i=1}^{n} \sum_{j=i+1}^{n} (1-\text{Similarity } (c_i, c_j)))/(\frac{n}{2} * (n-1))$.

The *pairs-F1* metrics [22] evaluates the correctness of sequence by using the F1 score of every pair of POIs in a sequence: $pairs - F1 =$

[1] http://www.yongliu.org/datasets/

| | Models | Precision Pair | Recall Pair | Pair-F1 | Diver-sity | Displa-cement | $NDCG_{10}$ |
|---|---|---|---|---|---|---|---|
| Weeplace Dataset | Popularity | 0.30000 | 0.16666 | 0.21428 | 1.20000 | 23.30785 | 0.2867 |
| | Apriori | 0.46079 | 0.23088 | 0.30762 | 1.90000 | 13.00000 | 0.2921 |
| | Markov | 0.49411 | 0.24711 | 0.32945 | 2.50000 | 11.72130 | 0.2979 |
| | HITS | 0.49981 | 0.27336 | 0.35342 | 4.00000 | 10.55233 | 0.3107 |
| | RNN | 0.49788 | 0.27618 | 0.35528 | 6.22000 | 10.27620 | 0.4536 |
| | LSTM | 0.51557 | 0.27500 | 0.35868 | 6.73000 | 10.00023 | 0.5633 |
| | CLoSe-RNN | 0.62422 | 0.41970 | 0.50192 | 7.09120 | 8.22990 | 0.5637 |
| | CLoSe-LSTM | 0.67771 | 0.43100 | **0.52690*** | **7.11820** | **7.77014** | **0.5771** |
| Gowalla Dataset | Popularity | 0.36442 | 0.20010 | 0.25834 | 3.20000 | 25.22877 | 0.2885 |
| | Apriori | 0.46922 | 0.24276 | 0.31997 | 3.33500 | 13.00000 | 0.2973 |
| | Markov | 0.49993 | 0.24981 | 0.33314 | 3.50000 | 11.22113 | 0.2989 |
| | HITS | 0.50653 | 0.27993 | 0.36058 | 4.00000 | 11.11224 | 0.3137 |
| | RNN | 0.51001 | 0.27896 | 0.36065 | 5.88441 | 11.00111 | 0.4566 |
| | LSTM | 0.53333 | 0.44000 | 0.48219 | 7.22533 | 10.33333 | 0.5683 |
| | CLoSe-RNN | 0.60914 | 0.43000 | 0.50412 | 8.44765 | 7.77669 | 0.5686 |
| | CLoSe-LSTM | 0.67112 | 0.44462 | **0.53487*** | **8.45001** | **7.71001** | **0.5791** |

**Table 1: Performance of models on different metrics**




**(a) Diversity in Weeplace** **(b) Diversity in Gowalla**

**(c) Displacement in Weeplace** **(d) Displacement in Gowalla**

**Figure 2: Diversity and displacement with sequence length**

$\frac{2*P_{PAIR}*R_{PAIR}}{P_{PAIR}+R_{PAIR}}$, where $P_{PAIR}$ and $R_{PAIR}$ are the precision and recall of the ordered POI pairs respectively. The *displacement* measures distance between predicted sequence ($seq_e$) and actual sequence ($seq_a$): Disp$(seq_a, seq_e) = \sum_{i=1}^{|seq_a|} \mid d(seq_{a_i}, seq_{e_i}) \mid$. We also use NDCG (normalized discounted cumulative gain) metrics.

**Experimental settings**:The daily check-in frequency of ≥10 was considered a valid sequence and users with <10 valid sequences were ignored. For each user, 10 most frequently checked-in places were taken as starting point and 10 sequences per starting point were generated. The average metrics on these sequences were observed. The Popularity and Apriori models used distance threshold of 2 K.m. The CLoSe-LSTM used 512 hidden states, the CLoSe-RNN used 5 layers and 256 nodes, input sequence length was 10, data was fed in mini batches of size 50, embedding vectors were of size 384, and the experiment was repeated for 100 epochs. The learning rate was 0.002 and the gradients were clipped at 5 to prevent overfitting.

**Evaluation Baselines:** We used following baselines: 1) *Popularity*: It selects the POI that has highest check-in frequency within a region, (2) *Markov Chain*: We use first order Markov Chain on Laplace smoothed state-transition matrix and initial probability matrix derived from the check-in data and personalized for each user, (3) *Apriori approach* [11, 12]: The most frequently checked-in place of a user is used as a starting point and places within a threshold distance ($\epsilon$) from it are used to get the candidate sets. The top-k trips with: (i) at most 8 hours travel time, (ii) higher trip score (calculated by preference scores, see Eqn. 3 and Eqn. 5), and (iii) lower travel time (see Eqn. 3) are selected, (4) *HITS approach* [29]: The locality preferences are incorporated by hierarchically organizing locations into regions and by computing hub scores of user and authority scores of places. The inference is made by using the adjacent matrix between users and locations with respect to the region. The score of a sequence is determined using the hub scores of visitors of the sequence and the authority scores of places weighted by the probability that people would consider the sequence.

**Experimental Results and Discussions:** The performance of different models is illustrated in Table 1 and Figure 2. The non-personalized models (Popularity and Apriori) performed low. The first-order Markov model outperformed Popularity and Apriori models which is due to the personalization implied from separate initial-probability and state-transition tables for each user. The HITS model outperformed Markov model. As it relies on segregation of places into regions and finding the authority and hub scores of the places and users within the regions, its performance depends on the region generation approach. We used a radius of 10 K.m. from a starting POI to generate such regions. Its performance with the radius of 5 K.m. and 15 K.m. was on par with Popularity model. The RNN slightly outperformed HITS model. This might be because of its capability to retain information of previous items from the sequence. The regular LSTM performed slightly better than RNN due to its implicit ability to cope with vanishing gradient problem. The **CLoSe-LSTM** outperformed CLoSe-RNN but was slower as it used multiple epochs and required lot of training time.

## 5 CONCLUSION AND FUTURE WORK

We formulated the contextual personalized POI sequence modeling by extending the RNN and LSTM. We incorporated different contexts (e.g., social, temporal, categorical, and spatial) into the hidden and output layer. We propagated the feature vector to all layers of the network and retained information that was valid throughout the sequence. We demonstrated the performance gain of proposed models using different metrics on two real-world datasets. In the future, we would like to add text attributes in our model.

## ACKNOWLEDGEMENT

This research is supported by U.S. Army Research Lab with grant no. W911NF-12-R-0012 and by FIU Dissertation Year Fellowship.

# REFERENCES

[1] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 363–372. ACM, 2013.

[2] Jia-Dong Zhang and Chi-Yin Chow. Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 443–452. ACM, 2015.

[3] Ramesh Baral and Tao Li. Exploiting the roles of aspects in personalized poi recommender systems. *Data Mining and Knowledge Discovery*, pages 1–24, 2017.

[4] Ramesh Baral and Tao Li. Maps: A multi aspect personalized poi recommender system. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 281–284. ACM, 2016.

[5] Ramesh Baral, Dingding Wang, Tao Li, and Shu-Ching Chen. Geotecs: exploiting geographical, temporal, categorical and social aspects for personalized poi recommendation. In *Information Reuse and Integration (IRI), 2016 IEEE 17th International Conference on*, pages 94–101. IEEE, 2016.

[6] Ramesh Baral, XiaoLong Zhu, SS Iyengar, and Tao Li. Reel: Review aware explanation of location recommendation. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, pages 23–32. ACM, 2018.

[7] Tomas Mikolov and Geoffrey Zweig. Context dependent recurrent neural network language model. *SLT*, 12:234–239, 2012.

[8] Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*, 2016.

[9] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[10] Chenyi Zhang, Hongwei Liang, Ke Wang, and Jianling Sun. Personalized trip recommendation with poi availability and uncertain traveling time. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 911–920. ACM, 2015.

[11] Zhiwen Yu, Huang Xu, Zhe Yang, and Bin Guo. Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. *IEEE Transactions on Human-Machine Systems*, 46(1):151–158, 2016.

[12] Eric Hsueh-Chan Lu, Ching-Yu Chen, and Vincent S Tseng. Personalized trip recommendation with multiple constraints by mining user check-in behaviors. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 209–218. ACM, 2012.

[13] Fabian Bohnert, Ingrid Zukerman, and Junaidy Laures. Geckommender: Personalised theme and tour recommendations for museums. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 26–37. Springer, 2012.

[14] Yong Ge, Qi Liu, Hui Xiong, Alexander Tuzhilin, and Jian Chen. Cost-aware travel tour recommendation. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 983–991. ACM, 2011.

[15] Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. Personalized travel package recommendation. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 407–416. IEEE, 2011.

[16] Qi Liu, Enhong Chen, Hui Xiong, Yong Ge, Zhongmou Li, and Xiang Wu. A cocktail approach for travel package recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 26(2):278–293, 2014.

[17] Chao Chen, Daqing Zhang, Bin Guo, Xiaojuan Ma, Gang Pan, and Zhaohui Wu. Tripplanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1259–1273, 2015.

[18] Kevin Meehan, Tom Lunney, Kevin Curran, and Aiden McCaughey. Context-aware intelligent recommendation system for tourism. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 328–331. IEEE, 2013.

[19] Munmun De Choudhury, Moran Feldman, Sihem Amer-Yahia, Nadav Golbandi, Ronny Lempel, and Cong Yu. Automatic construction of travel itineraries using social breadcrumbs. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pages 35–44. ACM, 2010.

[20] Shuhui Jiang, Xueming Qian, Tao Mei, and Yun Fu. Personalized travel sequence recommendation on multi-source big social media. *IEEE Transactions on Big Data*, 2(1):43–56, 2016.

[21] Xiaoting Wang, Christopher Leckie, Jeffrey Chan, Kwan Hui Lim, and Tharshan Vaithianathan. Improving personalized trip recommendation by avoiding crowds. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 25–34. ACM, 2016.

[22] Dawei Chen, Cheng Soon Ong, and Lexing Xie. Learning points and routes to recommend trajectories. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2227–2232. ACM, 2016.

[23] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems*, pages 1–32, 2017.

[24] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. Recurrent recommender networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 495–503. ACM, 2017.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[26] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[27] Ronald J Williams and David Zipser. Gradient-based learning algorithms for recurrent networks and their computational complexity. *Backpropagation: Theory, architectures, and applications*, 1:433–486, 1995.

[28] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao. Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 733–738. ACM, 2013.

[29] Yu Zheng and Xing Xie. Learning travel recommendations from user-generated gps traces. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):2, 2011.