A Contextual Attention Recurrent Architecture for Context-Aware Venue Recommendation

Jarana Manotumruksa
University of Glasgow
Glasgow, Scotland, UK
j.manotumruksa.1@research.gla.ac.uk

Craig Macdonald, Iadh Ounis
University of Glasgow
Glasgow, Scotland, UK
first.lastname@glasgow.ac.uk

ABSTRACT

Venue recommendation systems aim to effectively rank a list of interesting venues users should visit based on their historical feedback (e.g. checkins). Such systems are increasingly deployed by Location-based Social Networks (LBSNs) such as Foursquare and Yelp to enhance their usefulness to users. Recently, various RNN architectures have been proposed to incorporate contextual information associated with the users' sequence of checkins (e.g. time of the day, location of venues) to effectively capture the users' dynamic preferences. However, these architectures assume that different types of contexts have an identical impact on the users' preferences, which may not hold in practice. For example, an *ordinary context* – such as the time of the day – reflects the user's current contextual preferences, whereas a *transition context* – such as a time interval from their last visited venue – indicates a transition effect from past behaviour to future behaviour. To address these challenges, we propose a novel Contextual Attention Recurrent Architecture (CARA) that leverages both sequences of feedback and contextual information associated with the sequences to capture the users' dynamic preferences. Our proposed recurrent architecture consists of two types of gating mechanisms, namely 1) a *contextual attention gate* that controls the influence of the *ordinary* context on the users' contextual preferences and 2) a *time- and geo-based gate* that controls the influence of the hidden state from the previous checkin based on the *transition* context. Thorough experiments on three large checkin and rating datasets from commercial LBSNs demonstrate the effectiveness of our proposed CARA architecture by significantly outperforming many state-of-the-art RNN architectures and factorisation approaches.

ACM Reference Format:
Jarana Manotumruksa and Craig Macdonald, Iadh Ounis. 2018. A Contextual Attention Recurrent Architecture for Context-Aware Venue Recommendation. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209978.3210042

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210042

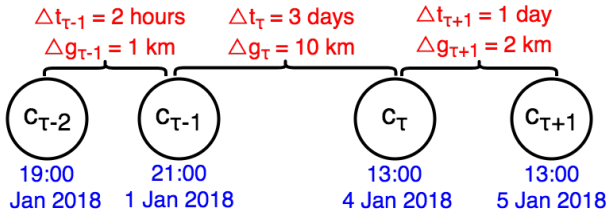


Figure 1: An illustration of the user's sequence of checkins, where each timestamp of the checkin is highlighted in blue, Δt and Δg are the time interval and the distance between checkins at time step τ, respectively (red text).

1 INTRODUCTION

Users in Location-Based Social Networks (LBSNs), such as Yelp and Foursquare, can share their location with their friends by making checkins at venues (e.g. museums, restaurants and shops) they have visited, resulting in huge amounts of user check-in data. Effective venue recommendation systems have become an essential application for LBSNs that facilitate users finding interesting venues based on their historical checkins. Collaborative filtering techniques such as Matrix Factorisation (MF) [17] are widely used to recommend a personalised ranked list of venues to the users. MF-based approaches typically aim to embed the users' and venues' preferences within *latent factors*, which are combined with a dot product operator to estimate the user's preference for a given venue. Approaches on MF typically encapsulate contextual information about the user, which can help to make effective recommendations for users with few historical checkins, known as the *cold-start* problem [22, 30, 32].

In recent years, various approaches have been proposed to leverage Deep Neural Network (DNN) algorithms for recommendation systems [3, 10, 11, 21, 28, 31]. Among various DNN techniques, the Recurrent Neural Network (RNN) models have been widely used to extend the MF-based approaches to capture users' short-term preferences from the users' sequence of observed feedback [1, 21, 26, 28, 31, 37]. Here, the short-term (*dynamic*) preferences assume that the next venue visited by a user is influenced by his/her recently visited venues (e.g. users may prefer to visit a bar directly after dinner at a restaurant).

A common technique to incorporate RNN models (e.g. *Long Short-Term Memory* (LSTM) units [13] and *Gated Recurrent Units* (GRU) [4]) into MF-based approaches is to feed a sequence of user-venue interactions/checkins into the recurrent models and use the hidden state of the recurrent models to represent the users' dynamic preferences [21, 28, 31, 35]. Next, the user's preference of a target venue is estimated by calculating the dot product of this representation of the user's dynamic preferences (i.e. the output of the recurrent models) and a latent factor of the target venue.

Although this technique can enhance the effectiveness of MF-based approaches, we argue that directly applying traditional RNN-based models to capture the users' dynamic preferences is not effective for Context-Aware Venue Recommendation (CAVR). In particular, the traditional RNN models are limited as they can only take the sequential order of checkins into account and cannot incorporate the contextual information associated with the checkins (e.g. timestamp of a user's checkin and the geographical location of the checkin). Indeed, such contexts have been shown to play an important role in producing effective CAVR recommendations [6, 22, 30, 32].

To address the above challenge, various approaches have been proposed to extend the RNN models to incorporate the contextual information of observed feedback into various recommendation settings excepting CAVR [1, 14, 19, 23, 26, 29, 37]. For example, Zhu *et al.* [37] proposed an extension of LSTM (TimeLSTM) by introducing time gates that control the influence of the hidden state of a previous LSTM unit based on the time interval between successive observed feedbacks. Indeed, they assume that the shorter the time interval between two successive feedback, the stronger the correlation between these two feedbacks and vice versa. However, their proposed model was designed for a particular type of contextual information (i.e. time intervals) and is not flexible to incorporate other types of context (e.g. distance between venues). We argue that the time gates proposed by Zhu *et al.* [37] are not effective to model the sequences of checkins in LBSNs. Figure 1 illustrates the user's sequential order of checkins. Let's consider the time intervals and distances between three successive checkins $c_{\tau-1}$, c_τ and $c_{\tau+1}$. With Zhu *et al.*'s time gates, checkin $c_{\tau-1}$ (c_τ) will have a small impact on checkin c_τ ($c_\tau + 1$) due to the long time interval between $c_{\tau-1}$ (c_τ) and c_τ ($c_\tau + 1$). This is counter-intuitive since checkin c_τ may have a strong impact on checkin $c_{\tau+1}$ due to the geographical distance between them. For example, a user may decide to visit a museum near the restaurant they had dinner at the previous day. Although the time interval from the previous checkin is long (> 24 hours), geographically, the restaurant and museum are close.

Recently, several works (e.g. CGRU [26] and LatentCross [1], for product and movie recommendation systems, respectively) have extended traditional RNN architectures for recommendation systems to incorporate different types of contextual information of the observed feedback sequences. However, we argue that their proposed architectures are limited for context-aware venue recommendation in several respects. In Figure 1, we highlight two types of contextual information associated with sequences of checkins, namely: the *ordinary* and *transition* contexts. The ordinary context represents the (absolute) timestamp and the geographical position of the checkin, while the transition context represents the (relative) time interval and distance between successive checkins. A disadvantage of the aforementioned RNN architectures is that they rely on a quantised mapping procedure (i.e. to convert continuous values of time intervals and distances to discrete features and represent these transition contexts using low-dimensional embedding vectors), which may result in a loss of granularity. In addition, their proposed architectures treat the ordinary and transition contexts dependently. However, we argue that these contexts influence the user's dynamic preference differently and should be considered independently. Indeed, the ordinary context reflects the user's contextual preference on a venue, while the transition context reflects the influence that one checkin has on its successor.

To address these challenges, we propose a Contextual Attention Recurrent Architecture (CARA) that leverages the sequential of users' checkins to model the users' dynamic preferences. In particular, our contributions are summarised below:

- We propose a Contextual Attention Recurrent Architecture (CARA) that independently incorporates different types of contextual information to model the users' dynamic preference for CAVR. Our proposed recurrent architecture differs from the recently proposed CGRU [26] and LatentCross [1] architectures in three aspects: (1) CARA includes gating mechanisms that control the influence of the hidden states between recurrent units, (2) CARA supports both discrete and continuous inputs and (3) CARA treats different types of context differently, In contrast, both CGRU and LatentCross do not support these features.
- Within the CARA architecture, we propose two gating mechanisms: a Contextual Attention Gate (CAG) and a Time- and Spatial-based Gate (TSG). The CAG controls the influence of context and previous visited venues, while TSG controls the influence of the hidden state of the previous RNN unit based on time interval and geographical distances between two successive checkins. Note that our proposed TSG differs from the time gates in TimeGRU [37] as we can incorporate multiple types of context, whereas TimeGRU supports only the time intervals. To the best of our knowledge, this work is the first that incorporates geographical information into an RNN architecture for CAVR.
- We conduct comprehensive experiments on 2 large-scale real-world datasets, from Brightkite and Foursquare, to demonstrate the effectiveness of our proposed CARA architecture for CAVR by comparing with state-of-the-art venue recommendation approaches. The experimental results demonstrate that CARA consistently and significantly outperforms various existing strong RNN models.

This paper is structured as follows: Section 2 provides a background in the literature on CAVR, as well as recent trends in applying Deep Neural Networks to recommendation systems; Section 3 details specific existing RNN-based recommendation architectures from the literature, and highlights 5 limitations in these approaches; Section 4 details our proposed CARA architecture that addresses all 5 limitations; Experimental setup and results are provided in Sections 5 & 6, respectively. Concluding remarks follow in Section 7.

2 BACKGROUND

Context-Aware Venue Recommendation (CAVR). Collaborative Filtering (CF) techniques such as Matrix Factorisation (MF) [17], Factorisation Machines [24] and Bayesian Personalised Ranking (BPR) [25] have been widely used in recommendation systems. Such factorisation-based approaches assume that users who have visited similar venues share similar preferences, and hence are likely to visit similar venues in the future. Previous works on venue recommendation have shown that the contextual information associated with the users' observed feedback (time of the day, location) play an important role to enhance the effectiveness of CAVR as well as to alleviate the cold-start problem [6, 7, 22, 30, 32, 34, 36]. For example, Yao *et al.* [30] extended the traditional MF-based approach by exploiting a high-order tensor instead of a traditional

user-venue matrix to model multi-dimensional contextual information. Manotumruksa *et al.* [22] and Yuan *et al.* [32] extended BPR to incorporate the geographical location of venues to alleviate the cold-start problem by sampling negative venues based on an assumption that users prefer nearby venues over distant ones. Zhao *et al.* [36] proposed Spatial-TEmporaL LAtent Ranking (STELLAR), which recommends a list of venues based on the user's context such as time and recent checkins.

Deep Neural Network Recommendation Systems. With the impressive successes of Deep Neural Network (DNN) models in domains such as speech recognition, computer vision and natural language processing (e.g. [9, 15, 33]), various approaches (e.g. [3, 10, 11, 18, 19, 21, 31]) have been proposed to exploit DNN models for recommendation systems. For example, He *et al.* [11] and Cheng *et al.* [3] proposed to exploit Multi Layer Perceptron (MLP) models to capture the complex structure of user-item interactions. An advantage of such MLP-based models is their ability to capture the user's complex structure using a DNN architecture and a non-linear function such as sigmoid. Liu *et al.* [18], Liu *et al.* [19] and Manotumruksa *et al.* [21] all exploited Recurrent Neural Networks (RNNs) to model the sequential order of the users' observed feedback. Due to the complex and overwhelming parameters of DNN models, such DNN-based CF approaches are prone to overfitting. Several empirical studies [10, 11, 27] have demonstrated that the use of generalised distillation techniques, such as dropout & regularisation, as well as pooling techniques can alleviate the overfitting problems inherent to DNN-based models. However, while the previous attempts mentioned above mainly focus on how to exploit DNN models to enhance the quality of recommendations, few attempts have focused on how to extend such DNN models to address particular challenges in recommendation systems. In this work, we propose to extend the traditional RNN architecture to incorporate the contextual information for CAVR. The next section describes the most recent work extensions of RNN for recommendation systems.

3 RECURRENT NEURAL NETWORK MODELS FOR RECOMMENDATION SYSTEMS

We first formalise the problem statement. Then, we briefly describe the MF-based approaches that exploit RNN models to model the sequential order of users' feedback (Section 3.2) and state-of-the-art recurrent architectures that take contextual information into account (Section 3.3). Note that these recurrent architectures were not originally proposed for CAVR but are sufficiently flexible to be applied to this task. For simplicity, we explain their proposed architectures in the context of venue recommendation and use a Gated Recurrent Unit (GRU) architecture [4] to explain their proposed architectures. Finally, Section 3.4 summarises the elicited limitations of these MF-based and RNN-based approaches. Later, in Section 4, we describe our proposed recurrent architecture that addresses these limitations.

3.1 Problem Statement

The task of context-aware venue recommendation is to generate a ranked list of venues that a user might visit given his/her preferred context and historical feedback (e.g. previously visited venues from checkin data). Let $c_{i,j,t} \in C$ denote a user $i \in \mathcal{U}$ who has checked-in into venue $j \in \mathcal{V}$ at timestamp t. Note that $c_{i,j,t} = 0$ means user i has not made a checkin at venue j at time t. Let $\mathcal{V}_i^+$ denote the list of venues that the user i has previously visited, sorted by time and let $\mathcal{S}_i$ denote the set of sequence of checkins (e.g. $\mathcal{S}_i = \{[c_1], [c_1, c_2], [c_1, c_2, c_3]\}$). $s_{i,t} = \left\{c = (i, j, \dot{t}) \in C \mid \dot{t} < t\right\} \subset \mathcal{S}_i$ denotes the sequence of checkins of user i up to time t. We use $s_{i,t}^{\tau}$ to denote the τ-th checkin in the sequence. t^{τ} denotes the timestamp of τ-th checkin. lat_j, lng_j are the latitude and longitude of checkin/venue j.

3.2 Recurrent-based Factorisation Approaches

Factorisation-based approaches aim to approximate matrix C by finding a decomposition of C into latent factors. For example, the predictions by an approach based on a Matrix Factorisation (MF) [17] are generally obtained from a dot product of latent factors of users $U \in \mathbb{R}^{|\mathcal{U}| \times d}$ and venues $V \in \mathbb{R}^{|\mathcal{V}| \times d}$ where d is the number of latent dimensions (i.e. $\hat{c}_{i,j} = \phi u_i^T \phi v_j$), and ϕu_i and ϕv_j are the latent factors of user i and venue j, respectively. Various approaches [21, 28, 31, 35] have been proposed to extend MF by exploiting Recurrent Neural Network (RNN) models to capture the user's dynamic preferences from the sequence of user's checkins. In particular, given the sequence of a user's checkins $S_{i,t}$, the output of a RNN model, h_{τ}, is used to represent a user's dynamic preferences and modify the MF-based approaches as follows:

$$\hat{c}_{i,j} = (\phi u_i + h_{\tau})^T \phi v_j \tag{1}$$

However, the operation that combines latent factors ($\phi u_i, \phi v_j$) and hidden state h_{τ} need not be limited to the dot product and summation. Previous works [10, 11, 21] have shown that using either element-wise product or concatenation operators between the latent factors and hidden state h_{τ}, with a non-linear function such as a sigmoid function, are more effective than the simple dot product operation in capturing the complex structure of user-venue interactions. In this paper, we argue that the current RNN-based factorisation approaches [21, 28, 31, 35] that exploit traditional RNN models to capture the users' dynamic preferences are not effective, because they only consider the sequence of previously visited venues and ignore the contextual information associated with the checkins (**Limitation 1**).

3.3 Gating Mechanisms of Recurrent Models

In this section, we discuss extensions of traditional RNN models proposed in previous works [1, 4, 26, 37]. Traditional RNN models usually suffer from the vanishing gradient problem when the models are trained from long sequences of observed checkins [4, 13]. Recurrent units such as Long-Short Term Memory (LSTM) [13] and Gated Recurrent Unit (GRU) [4] are extensions of traditional RNN models that use gating mechanisms to control the influence of a hidden state of previous step, $h_{\tau-1}$.

3.3.1 Gated Recurrent Units. To alleviate the gradient problem, Chung *et al.* [4] proposed a variant of RNN models, Gated Recurrent Units (GRU), which consists of gating mechanisms that control the influence of the hidden state of previous unit $h_{\tau-1}$ in the current unit at time step τ. Indeed, GRU can learn to ignore the previous units if necessary, whereas traditional RNN models cannot. In particular, given the user's sequence of checkins $s_{i,t}$ and the user's dynamic preference at time step τ, the hidden state, h_{τ}, is estimated using the gating mechanisms, which are defined as:

$$[z_{\tau}, r_{\tau}] = \sigma(W \phi v_j^{\tau} + R h_{\tau-1} + b) \tag{2}$$

$$\widetilde{h}_\tau = \tanh(W\phi v_j^\tau + R(r_\tau \odot h_{\tau-1})) \quad (3)$$

$$h_\tau = (1 - z_\tau)h_{\tau-1} + z_\tau \widetilde{h}_\tau \quad (4)$$

where z_τ, r_τ are update and reset gates, respectively. $\widetilde{h}_\tau$ is a candidate hidden state, ϕv_j^τ is the latent factor of the venue j that user i visited at time step τ (i.e. $s_{i,t}^\tau$). $\sigma()$ and tanh() are the sigmoid and hyperbolic tangent functions, respectively. R is a recurrent connection weight matrix that captures sequential signals between every two adjacent hidden states h_τ and $h_{\tau-1}$, using $\odot$, which denotes the element-wise product. Finally, W, b are, respectively, the transition matrix between the latent factors of venues, and the corresponding bias. We note that $\theta_r = \{W, R, b\}$ denotes the set of parameters of the GRU units. The advantage of GRU over the traditional RNN models is the ability to control the influence of the hidden state of previous step $h_{\tau-1}$ based on the reset and update gates z_τ, r_τ as well as the candidate hidden state $\widetilde{h}_\tau$ (see Equation (4)).

From now on, we explain the recurrent architectures proposed in recent works [1, 26, 37] in terms of the GRU architecture, due to its relative simplicity (i.e. less parameters compared to LSTM). It is of note that none of these architectures were originally proposed for CAVR but are sufficiently flexible to be applied to the CAVR task. However, for reasons of uniformity, we explain all of the following approaches in terms of the CAVR task, thereby replacing item with venue, etc.

3.3.2 TimeGRU. While the GRU architecture can alleviate the vanishing gradient problem, it cannot leverage contextual information associated with the checkins. Zhu *et al.* [37] proposed to extend the GRU units to incorporate the time interval (i.e. the transition contexts) between successive checkins [1]. The left box of Figure 2 illustrate their proposed GRU units. In particular, they modify the candidate hidden state $\widetilde{h}_\tau$ (Equation (3)) with their proposed time gate T_τ, which is defined as:

$$T_\tau = \sigma_t(W\phi v_j^\tau + \sigma(\Delta t_\tau W_t) + b \quad (5)$$

$$\widetilde{h}_\tau = \tanh(W\phi v_j^\tau + R(r_\tau \odot T_\tau \odot h_{\tau-1}) + b) \quad (6)$$

where $\Delta t_\tau = t^\tau - t^{\tau-1}$ is the time interval between checkins $s_{i,t}^\tau$ and $s_{i,t}^{\tau-1}$. t^τ captures the correlation between the current venue v_j^τ and the time interval Δt^τ. Then, the time gate T_τ is used to control the influence of previous hidden state $h_{\tau-1}$ in Equation (6). In particular, the previous hidden state $h_{\tau-1}$ is not only controlled by the reset gate r_τ but also by their proposed time gate T_τ. We argue that there are two limitations that arise. First, TimeGRU can only incorporate the transition context (i.e. the time intervals between successive checkins, Δt_τ) but not the current context of the user, (i.e. the ordinary context, such as the time of the day when the user makes a checkin) (**Limitation 2**). Second, their proposed time gate is not sufficiently flexible to incorporate different types of transition context associated with the checkins such as the geographical distance between two successive checkins (**Limitation 3**).

[1] Although Zhu *et al.* [37] used the LSTM architecture to explain their proposed recurrent units, they claimed that their proposed architecture is sufficiently flexible to apply to a GRU architecture.

3.3.3 Context-aware GRU architectures. To address **Limitation 2**, Smirnova and Vasile [26] proposed a Contextual RNN architecture that can incorporate both the transition and ordinary context of observed checkins[2]. Their contributions were two fold: context-dependent venue representations and contextual GRU units. As shown in the second box in Figure 2, they proposed a concatenation integration function to model context-dependent venue representations. In particular, at a given time step τ, the input of the GRU unit is the concatenation of the latent factors of the ordinary and transition contexts as well as the latent factors of the venue. Since both the ordinary and transition contexts for the time dimension are continuous values (e.g. the timestamp t^τ, time interval Δt_τ and geographical distance Δg_τ), previous works [1, 14, 26, 36] have relied on mapping approaches to represent such context. For example, the ordinary context such as timestamp t^τ can be split into discrete features - month, hour of the day and day of the week. Next, 12, 24 and 7 bits are used to represent the month, hour and day, respectively, and convert the binary code into a unique decimal digit as a timestamp id. Similarly, the transition context - e.g. as the time interval Δt^τ can be quantised as the time interval id using the following function $ind(\Delta t^\tau) = \lceil \frac{\Delta t^\tau}{\delta T} \rceil$, where δT is a 1-hour interval. This technique can be similarly applied to quantise the geographical distance Δg_τ. Then, the timestamp t^τ, the time interval Δt^τ and the geographical distance Δg_τ can be represented as latent factors of time, time interval and distance, $\phi t^\tau, \phi \Delta t_\tau, \phi \Delta g_\tau \in \mathcal{R}^d$, respectively.

Next, Smirnova and Vasile [26] extended the transition matrix W of the GRU unit to be context-dependent, thereby aiming to capture the users' dynamic contextual preferences. In particular, they introduce the contextual matrix U, to condition the transition matrix W of a GRU unit as follows:

$$\begin{aligned} z_\tau &= \sigma(Wx^\tau \odot \underline{U_u xc^\tau}) + R_u h_{\tau-1} \\ r_\tau &= \sigma(Wx^\tau \odot U_r xc^\tau) + R_r h_{\tau-1} \\ \widetilde{h}_\tau &= \sigma(Wx^\tau + R_h(r_\tau \odot h_{\tau-1}) \odot U_h xc^\tau) \end{aligned} \quad (7)$$

where $x^\tau = [\phi v_j^\tau; \phi t^\tau; \phi \Delta t^\tau; \phi \Delta g_\tau]$ and $xc^\tau = [\phi t^\tau; \phi \Delta t^\tau; \phi \Delta g_\tau]$ are their proposed context-dependent venue and context representations, respectively. Recently, building upon Smirnova and Vasile's work [26], Beutel *et al.* [1] explored various approaches to effectively incorporate the latent factors of context xc^τ into RNN models. They proposed LatentCross, a technique that incorporates contextual information in the GRU, by performing an element-wise product of the latent factors of context xc^τ with the model's hidden states h_τ. The third box in Figure 2 illustrates how LatentCross works. The inputs of the GRU unit are the concatenation of all latent factors x^τ (black line) and the concatenation of latent factors of context xc^τ (red line). In particular, they modify Equation (4) with the latent factors of context, xc_τ, as follows:

$$h_\tau = (1 + xc^\tau) \odot [(1 - z_\tau)h_{\tau-1} + z_\tau \widetilde{h}_\tau] \quad (8)$$

Note that both CGRU and LatentCross are the most recent works that explore various techniques to incorporate context into recurrent models. However, we argue that there are two limitations in their proposed GRU architectures. First, their proposed architectures treat the ordinary and transition context similarly. We argue

[2] Although proposed and evaluated in the context of e-commerce item recommendation, recall that we explain this approach in the context of venue recommendation.

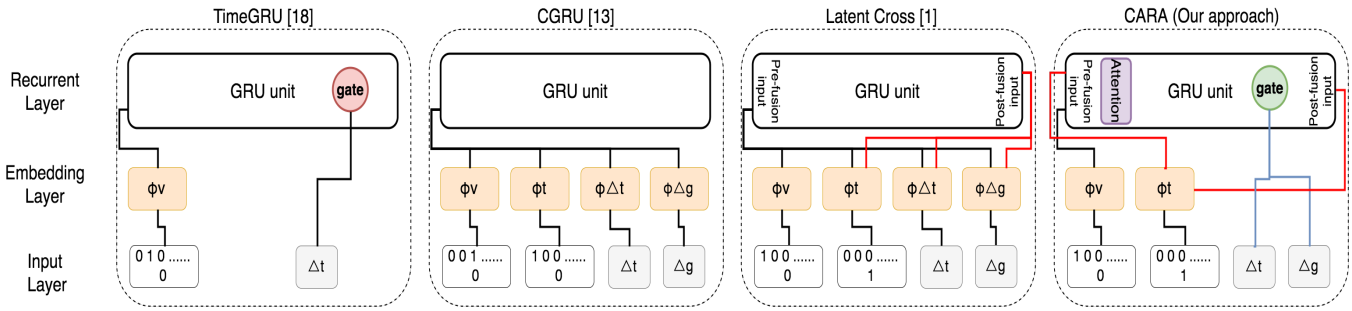


Figure 2: Diagrams of existing recurrent architectures and our proposed Contextual Attention Recurrent Architecture (CARA).

that different types of context might influence the user's dynamic preferences differently (**Limitation 4**). For example, the ordinary context should influence the user's contextual preference on a current visited venue, while the transition context should influence the correlation between the current and previously visited venues. Second, there is a loss of granularity from the quantisation mapping functions used to represent the transition context (**Limitation 5**).

3.4 Summary of Limitations

To conclude, in the above analysis, we have identified five limitations of RNN-based models from the literature [1, 26, 28, 31, 37]:
Limitation 1: There is an inherent disadvantage in the traditional RNN models that model the user's sequential order of checkins by leveraging only the sequence of previously visited venues and ignoring the context associated with the checkins.
Limitation 2: The GRU architecture for which this limitation applies (TimeGRU [26]) can only incorporate transition contexts.
Limitation 3: The time gating mechanism proposed by Zhu *et al.* [37] is not sufficiently flexible to incorporate different types of context.
Limitation 4: GRU architectures for which this limitation applies (CGRU [26] and LatentCross [1]) treat the ordinary and transition context similarly.
Limitation 5: There is an inherent disadvantage in the GRU architectures (CGRU [26] and LatentCross [1]) that rely on the quantised mapping procedures to represent the transition context.

Next, we describe our proposed Contextual Attention Recurrent Architecture (CARA), which addresses all of the elicited limitations.

4 CONTEXTUAL ATTENTION RECURRENT ARCHITECTURE (CARA)

We propose a novel Contextual Attention Recurrent Architecture (CARA) for context-aware venue recommendation that effectively incorporates different types of contextual information from sequential feedback to model users' short-term preferences (Section 4.2). The proposed recurrent architecture consists of two types of gating mechanisms: namely Contextual Attention Gate (CAG) as well as Temporal and Spatial Gates (TSG), which are described in Section 4.2 and Section 4.3, respectively. In particular, our proposed recurrent architecture with contextual gates aims to address all **Limitation 1-5**. Later, in Section 6, we evaluate the effectiveness of our proposed recurrent architecture in comparison with various state-of-the-art context-aware RNN models.

4.1 Proposed Recurrent Architecture for Context-aware Venue Recommendation

Our proposed CARA architecture is illustrated in the rightmost box of Figure 2. The architecture consists of 4 layers: namely input, embedding, recurrent and output layers. In particular, CARA aims to generate the ranked-list of venues that a user might prefer to visit at time t based on the sequences of checkins $s_{u,t}$. To address **Limitation 1**, in the input layer, at time step τ, given a user i, venue j and time t^τ, we compute the time interval and geographical distance between the given venue j and venue k previously visited at time step $\tau-1$, as $\Delta t^\tau = t^\tau - t^{\tau-1}$ and $\Delta g_\tau = dist(lat_j, lng_j, lat_k, lng_k)$, respectively. $dist()$ is the Haversine distance function. In the embedding layer, the latent factors of the user $\phi u_i \in U$, venue $\phi v_i^\tau \in Q$ and time $\phi t^\tau \in M$ are generated. $\theta_e = \{U, V, M\}$ denotes the set of parameters of the embedding layer. Note that we only consider the time of checkins as the ordinary context but our proposed architecture is flexible to support multiple types of ordinary context (e.g. current weather of the day).

Next, the latent factors of venue ϕv_j^τ, the latent factors of the given time ϕt^τ and the contextual transition features Δg_τ and Δt^τ are passed to the recurrent layer. The output of the recurrent layer is the hidden state of the recurrent unit at time step τ, h_τ, which is defined as follows:

$$h_\tau = f(\phi v_j^\tau, \phi t^\tau, \Delta t_\tau, \Delta g_\tau; \theta_r) \tag{9}$$

where $\theta_r = \{W, R, U, b\}$ denotes the set of parameters of the recurrent layer. More details on the recurrent units in the recurrent layer that generates the hidden state h_τ are described in Section 4.2 and Section 4.3 (Equations (13) - (18)). Finally, in the output layer, we estimate the preference of user i on venue j at timestamp t as follows:

$$\hat{c}_{i,j,t} = \phi u_i^T h_\tau \tag{10}$$

where $h_\tau \in \mathcal{R}^d$ is the hidden state of the recurrent layer. Previous works [1, 26, 37] have followed the *pointwise* paradigm, by using the softmax function to estimate the probability distribution over all venues given the hidden state h_τ and update the parameter based on the cross entropy loss (i.e. classification loss). However, others have shown that *pairwise* ranking losses result in more effective learning than those based on classification loss [2, 20–22, 25]. Therefore, we apply the pairwise Bayesian Personalised Ranking (BPR) [25] to learn the parameters $\Theta = \{\theta_e, \theta_r\}$, as follows:

$$\mathcal{J}(\Theta) = \sum_{i \in \mathcal{U}} \sum_{s_{i,t} \in S_i} \sum_{(i,j,t) \in s_{i,t}} \sum_{k \in \mathcal{V} - s_{i,t}} \log(\sigma(\hat{c}_{i,j,t} - \hat{c}_{i,k,t})) \tag{11}$$

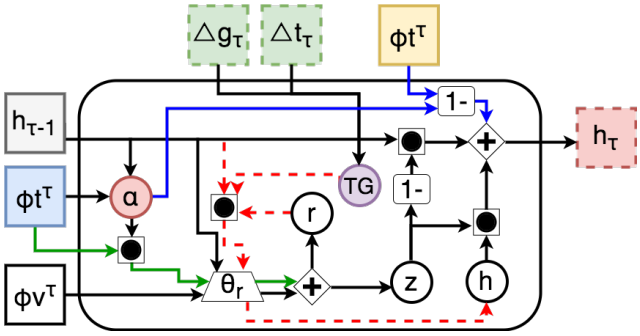


Figure 3: Our proposed Contextual Attention Recurrent Architecture (CARA). The Rectangle symbols indicate inputs of the unit, a red-dashed rectangle symbol indicates the output of the unit and the circle symbols are the units' gates.

4.2 Contextual Attention Gate (CAG)

We now describe how we extend the traditional Gated Recurrent Unit (GRU) to incorporate the ordinary context associated with the observed checkins in order to address **Limitation 2**. In particular, we further describe how to calculate the hidden state h_τ in Equation (9). Inspired by [8], we propose the Contextual Attention Gate (CAG), $\alpha \in \mathcal{R}^d$, which controls the influences of the latent factor of time δt^τ at each state as follows:

$$\alpha_\tau = \sigma(W_{\alpha,h} h_{\tau-1} + W_{\alpha,t} \phi t^\tau + b_\alpha) \quad (12)$$

The attention gate α_τ (red circle in Figure 3) aims to capture the correlation between the latent factor ϕt^τ at current step τ and the hidden state $h_{\tau-1}$ of previous step. Our proposed attention gate aims to capture the influence of the user's dynamic preferences $h_{\tau-1}$ on the current context t^τ. Then, we modify Equations (2)-(4) with the attention gate α_τ as follows:

$$[z_\tau, r_\tau] = \sigma(W \phi v_j^\tau + R h_{\tau-1} + W(\alpha_\tau \odot \phi t^\tau) + b) \quad (13)$$

$$\widetilde{h}_\tau = \tanh(W \phi v_j^\tau + R(r_\tau \odot h_{\tau-1}) + W(\alpha_\tau \odot \phi t^\tau) + b) \quad (14)$$

$$h_\tau = (1 + (1 - \alpha_\tau) \odot \phi t^\tau) \odot [(1 - z_\tau) h_{\tau-1} + z_\tau \widetilde{h}_\tau] \quad (15)$$

Unlike previous works [1, 26] that combine the latent factors of the venues and time using the concatenation operation, i.e. $[v^\tau; \phi t^\tau]$ (see Section 3.3.3), we argue that these two latent factors should be treated independently. Ideally, the ordinary context associated with checkins represents the user's contextual preferences abotu the venue, while the latent factors of the venues represent characteristics of the venues. Indeed, we can include the ordinary context, e.g. the latent factor of time ϕt^τ, into the GRU units in two ways: namely at the beginning and the end of the GRU unit. Cui *et al.* [5] described the inclusion of context features before the GRU unit as *pre-fusion* (blue box in Figure 3), and the inclusion of context features after the GRU unit as *post-fusion* (yellow box in Figure 3). In particular, by including the latent factor of time t^τ through pre-fusion (Equations (13) & (14), t^τ will affect the update of the hidden state of the current GRU unit though the update and reset gates z_τ, r_τ as well as the candidate hidden state $\widetilde{h}_\tau$. However, by including the latent factor of time ϕt^τ through post-fusion (Equation (15)), t^τ have more effect on the hidden state h_τ, the output of the GRU unit, and hence affects the next hidden state of next step $h_\tau + 1$.

Our proposed attention gate α_τ controls the influence of the latent factor of time t^τ on pre- and post- fusion. In particular, to address **Limitation 4**, our proposed CARA architecture uses a CAG gate to model the ordinary context and use TSG gates to model the transition context, which is described in the next section.

4.3 Time-and Spatial-based Gates (TSG)

In the previous section, we explained how to extend the GRU units to incorporate the ordinary context associated with observed checkins. As mentioned in Section 1, to effectively model the users' sequential order of checkins, we need to take the transition context into account. In this section, we describe how to extend the GRU units to incorporate the transition context such as the time intervals and the geographical distances between successive checkins. The green-dashed boxes and purple circle in Figure 3 illustrate our proposed Time- and Spatial-based Gates (TSG). To address **Limitations 3 & 5**, inspired by the time gates proposed Zhu *et al.* [37], we propose to extend their time gate to incorporate the geographical distance between two checkins, Δg_τ, as follows:

$$T_\tau = \sigma_t(W_{tx} \phi v^\tau + \sigma(\Delta t_\tau W_t) + b_t) \quad (16)$$

$$G_\tau = \sigma_t(W_{gx} \phi v^\tau + \sigma(\Delta g_\tau W_g + b_g)) \quad (17)$$

where Δt^τ and Δg_τ are time interval and distances between checkins c_τ and $c_{\tau-1}$, respectively. Note that unlike previous works [1, 18, 19, 26], our proposed TSG gates support using continuous values for a transition context, hence they do not rely on the quantised mapping procedure to represent a transition context. Then, we propose to combine these two gates using the element-wise product $TG_\tau = T_\tau \odot G_\tau$ and modify Equation (14) as follows:

$$\widetilde{h}_\tau = \tanh(W \phi v_j^\tau + R(r_\tau \odot TG_\tau \odot h_{\tau-1}) + W(\alpha_\tau \odot \phi t^\tau) + b) \quad (18)$$

The TG_τ gate and the reset gate r_τ together control the influence of the hidden state of previous step $h_{\tau-1}$. Unlike the time gate proposed by Zhu *et al.* [37], the TG_τ gate takes both the time intervals and the geographical distance of two successive checkins into account. Hence, even if the time interval between two checkins is long, the influence of the hidden state $h_{\tau-1}$ may not be decreased if the distance between the two checkins is short, based on the assumption we mentioned in Section 1. Later in Section 6, we compare the effectiveness of our proposed TSG gate in comparison with the time gate approach proposed by Zhu *et al.* [37].

5 EXPERIMENTAL SETUP

In this section, we evaluate the effectiveness of our proposed Contextual Attention Recurrent Architecture (CARA) in comparison with state-of-the-art recurrent models. In particular, to address **Limitations 1 - 5**, we address the following research questions:

- RQ1 *Can we enhance the effectiveness of traditional recurrent architecture by leveraging the ordinary and transition contexts associated with the sequence of checkins?*
- RQ2 *Is it important to model ordinary and transition contexts separately?*
- RQ3 *Does the use of the absolute continuous values of the transition context preserve the influence of successive checkins?*

Furthermore, as discussed in Section 3.4, no previous work has proposed a gating mechanism that can incorporate multiple types of

Table 1: Statistics of the three used datasets.

	Brightkite	Foursquare	Yelp
Number of normal users	14,374	10,766	38,945
Number of venues	5,050	10,695	34,245
Number of ratings or checkins	681,024	1,336,278	981,379
Number of cold-start users	5,578	154	6903
% density of User-Venue matrix	0.93	1.16	0.07

transition contexts such as time-base context and the geographical information of venues. Hence, our final research question:

RQ4 *Can our proposed Time- and Spatial-based Gates (TSG) that leverages multiple types of transition contexts (i.e. the time intervals and geographical distances between successive checkins) enhance the effectiveness of traditional recurrent units in capturing the user's dynamic preferences?*

5.1 Datasets & Measures

We conduct experiments using three publicly available large-scale LBSN checkin datasets. In particular, to show the generalisation of our proposed architecture across multiple LBSN platforms and sources of feedback evidence, we use two checkin datasets from Brightkite[3] and Foursquare[4], and a rating dataset from Yelp[5]. We follow the common practice from previous works [11, 21, 25] to remove venues with less than 10 checkins. Table 1 summarises the statistics of the filtered datasets. To evaluate the effectiveness of our proposed CARA architecture and following previous works [11, 21, 25], we adopt a *leave-one-out* evaluation methodology: for each user, we select their most recent checkin as a ground truth and randomly select 100 venues that they have not visited before as the testing set, where the remaining checkins are used as the training set. The context-aware venue recommendation task is thus to rank those 101 venues for each user given their preferred context (i.e. time), aiming to rank highest the recent, ground truth checkin. We conduct two separate experiments, namely: *Normal Users* (those with ≥ 10 checkins) and *Cold-start Users* (< 10 checkins) to evaluate the effectiveness of our proposed CARA architecture in the general and cold-start settings. Recommendation effectiveness is measured in terms of Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) on the ranked lists of venues – as applied in previous literature [11, 21, 31]. In particular, HR considers the ranking nature of the task, by taking into account the rank(s) of the venues that each user has previously visited/rated in the produced ranking, while NDCG goes further by considering the checkin frequency/rating value of the user as the graded relevance label. Finally, significance tests use a paired t-test.

5.2 Baselines

We compare our proposed Contextual Attention Recurrent Architecture (CARA) with various baselines, which can be categorised as the state-of-the-art RNN architectures and other factorisation-based approaches. As mentioned before that some approaches and frameworks may not be originally proposed for CAVR but are sufficiently flexible to be applied to this task without any disadvantages. We implement all baselines and our proposed approach using Keras[6], a deep learning framework built on top of Theano[7]. Our implementations are released as open source[8]. The choice of recurrent models is fixed to the GRU units proposed by Zhang *et al.* [4]. Table 2 distinguishes various baselines into different aspects as well as indicate their limitations mentioned in Section 3.4. The summary of the baselines are described below:

5.2.1 *Recurrent Neural Network Architectures.*

RNN. A traditional recurrent architecture proposed by Zhang *et al.* [35] that only takes the sequence of venues into account and ignores any contextual information associated with the checkins, used by [21, 28, 31] (see Section 3.3.1).

STGRU. A Spatial and Temporal recurrent model proposed by Liu *et al.* [19] that incorporates the transition context (i.e. the time intervals and distance between checkins) (see Section 2).

CAGRU. An extension of STGRU proposed by Liu *et al.* [18], which can incorporate both the ordinary and transition contexts (see Section 2).

TimeGRU. An extension of the GRU architecture that includes the time gate to incorporate the time interval between successive checkins. It was proposed by Zhu *et al.* [37] (see Section 3.3.2).

CGRU. An extension of the GRU architecture that can incorporate multiple types of context. It was proposed by Smirnova and Vasile [26] (see Section 3.3.3).

LatentCross. An extension of CGRU that supports pre and post fusion inputs. It was proposed by Beute *et al.* [1] (see Section 3.3.3).

5.2.2 *Factorisation Approaches.*

MF. The traditional matrix factorisation proposed by Koren *et al.* [17] that aims to accurately predict the users' checkin on the unvisited venues.

BPR. The classical pairwise ranking approach, coupled with matrix factorisation for user-venue checkin prediction, proposed by Rendle *et al.* [25].

GeoBPR. An extension of BPR that incorporate geographical location of venues to sample negative venues that are far away from the user's previous visits. It was proposed by Yuan *et al.* [32].

STELLAR. A Spatial-TEmporaL LAtent Ranking framework for CAVR that aims to recommend the list of venues based on the user's preferred time and last successive visits. It was proposed by Zhao *et al.* [36]. Note that this is the only context-aware framework that does not rely on the RNN-based approaches to model the users' sequential order of checkins.

NeuMF. A Neural Matrix Factorisation framework[9], proposed by He *et al.* [11], which exploits Multi-Level Perceptron (MLP) models to capture the complex structure of user-item interactions.

DRCF. A Deep Recurrent Collaborative Filtering framework, proposed by Manotumruksa *et al.* [21], which extends NeuMF [11] to exploit the traditional RNN to model the sequential order of users' checkins. DRCF consists of two components, with each component

[3] https://snap.stanford.edu/data/
[4] https://archive.org/details/201309_foursquare_dataset_umn
[5] https://www.yelp.com/dataset_challenge
[6] https://github.com/fchollet/keras
[7] http://deeplearning.net/software/theano
[8] https://github.com/feay1234/CARA
[9] https://github.com/hexiangnan/neural_collaborative_filtering

Table 2: Summary of factorisation-based approaches and Gated Recurrent Unit architectures.

	Factorisation Approaches						Recurrent Neural Network Architectures						
	MF [17]	BPR [25]	GeoBPR [32]	STELLAR [36]	NeuMF [11]	DRCF [21]	RNN [35]	STGRU [19]	CAGRU [18]	TimeGRU [37]	CGRU [26]	LatentCross [1]	CARA
Neural networks	×	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sequential-based	×	×	×	✓	×	✓	✓	✓	✓	✓	✓	✓	✓
Context-aware	×	×	only geo	✓	×	×	×	✓	✓	only time	✓	✓	✓
Ordinary/Transition	×	×	×	✓	×	×	×	only transition	✓	only transition	✓	✓	✓
Special gates	×	×	×	×	×	×	×	×	×	✓	×	×	✓
Limitations	-	-	-	-	-	-	1-5	2-5	2,4,5	4,5	4,5	4,5	-

having its own recurrent layer. Hence, to permit a fair comparison, we only compare CARA with its best-performing component, GMRF, which uses an element-wise product to combine the latent factors and the hidden units of the RNN model [21, Section 4.2]).

5.3 Recommendation Parameter Setup

Following [11, 21], we set the dimension of the latent factors d and hidden layers h_τ of our proposed CARA architecture and all of the RNN-based approaches to be identical: $d = 10$ across both datasets. Following He *et al.* [11], we randomly initialise all embeddings and recurrent layers' parameters, $\theta_r, \theta_e, \theta_h$, with a Gaussian distribution (with a mean of 0 and standard deviation of 0.01) and apply the mini-batch Adam optimiser [16] to optimise those parameters, which yields faster convergence than SGD and automatically adjusts the learning rate for each iteration. We initially set the learning rate to 0.001[10] and set the batch size to 256. As the impact of the recurrent parameters such as the size of the hidden state, have been explored in previous work [11, 12, 27], we omit varying the size of the hidden layers and the dimension of the latent factors in this work. Indeed, larger sizes of hidden layers and dimensions may cause overfitting and degrade the generalisation of the models [11, 12, 27].

Table 3: Performance in terms of HR@10 and NDCG@10 between various approaches. The best performing result is highlighted in bold; − and ∗ denote a significant difference compared to the best performing result, according to the paired t-test for $p < 0.05$ and $p < 0.01$, respectively.

	Brightkite		Foursquare		Yelp	
Model	HR	NDCG	HR	NCDG	HR	NDCG
Normal Users						
RNN	0.6657*	0.4407*	0.8302*	0.5762*	0.4164*	0.2146*
TimeGRU	0.7005*	0.4816*	0.8570*	0.6167*	0.4342*	0.2240*
STGRU	0.6888*	0.5493*	0.8496*	0.6865*	0.4254*	0.2365*
CAGRU	0.7180*	0.5545*	0.8498*	0.6474*	0.3799*	0.1989*
CGRU	0.6969*	0.5659*	0.8592*	0.6985*	0.5194*	0.3005*
LatentCross	0.7063*	0.5727*	0.8616*	0.6964*	0.5210*	0.2991*
CARA	**0.7385**	**0.6040**	**0.8851**	**0.7154**	**0.5587**	**0.3272**
Cold-Start Users						
RNN	0.6959*	0.4550*	0.8247	0.5260*	0.2420*	0.4540*
TimeGRU	0.7314*	0.5071*	0.8182	0.5788*	0.2398*	0.4592*
STGRU	0.7081*	0.5686*	0.7273*	0.5722*	0.2543*	0.4404*
CAGRU	0.7628	0.6035*	0.8377	0.6353	0.2205*	0.4055*
CGRU	0.7054*	0.5788*	0.7662*	0.5996-	0.3325*	0.5524*
LatentCross	0.7108*	0.5811*	0.8052*	**0.6600**	0.3223*	0.5398*
CARA	**0.7648**	**0.622**	**0.8636**	0.6505	**0.3493**	**0.5748**

6 EXPERIMENTAL RESULTS

Table 3 reports the effectiveness of various state-of-the-art GRU recommendation architectures, in terms of HR@10 and NDCG@10 on the three used datasets. Similarly, Table 4 reports the performance of our proposed CARA architecture in comparison with various factorisation approaches (as described in Section 5.2.2). Both tables contain two groups of rows, which report the effectiveness of various approaches under the *Normal Users* and *Cold-Start Users* experiments, respectively.

Firstly, on inspection of our reimplementations of the state-of-the-art GRU baselines in Table 3, we note that the relative venue recommendation quality of the baselines on the three datasets in terms of both HR and NDCG are consistent with the results reported for the various baselines in the corresponding literature [1, 18, 19, 26, 35, 37]. For instance, the extensions of the GRU architecture that incorporate the contextual information (LatentCross, CGRU, CAGRU, STGRU and TimeGRU) outperforms RNN across three datasets. Similarly, among the factorisation baselines in Table 4, we also observe the relative improvements of GeoBPR, STELLAR, NeuMF and DRCF compared to MF and BPR across the three datasets. While previous works (e.g. [1, 11, 26, 37]) used different datasets, our reimplementations of their approaches obtain similar relative improvements.

Comparing CARA with various GRU architectures in Table 3 on the Normal Users experiment, we observe that CARA consistently and significantly outperforms all the GRU baselines, for HR and NDCG, across all datasets. In particular, CARA improves NDCG by 5.47-8.93% and 2.42-10.50% over the recently proposed GRU architectures CAGRU, CGRU and LatentCross, for the Brightkite and Foursquare checkin datasets, respectively. These results imply that our proposed CARA architecture with Contextual Attention Gate (CAG) and Time-and Spatial-based Gate (TSG) is more effective than the state-of-the-art GRU architectures in modelling the sequences of users' checkins. Within the second groups of rows in Table 3, we further analyse the effectiveness of our proposed CARA architecture by comparing with the GRU baselines in the *Cold-Start Users* experiment. Similar to the results observed from the *Normal Users* experiment, CARA consistently and significantly outperforms all GRU baselines across three datasets in terms of HR and NDCG, except for NDCG on the Foursquare dataset, where LatentCross is statistically indistinguishable from CARA (difference in HR < 1.5%). Next, we note that unlike the Brightkite and Foursquare checkin datasets, the Yelp dataset consists of only user-venue ratings, and hence the sequential properties of visits to venues cannot be observed. Consequently, on both normal and cold-start user experiments, the performances of several GRU baselines (TimeGRU, STGRU and CAGRU) that consider the contextual information of the ratings are as effective as the RNN baseline that only considers the sequence of the user's ratings. In contrast, our proposed CARA architecture, which controls the influence of previous ratings based on both the time interval and geographical distance, is still the most effective across the different types of datasets. Overall, in response to research question RQ1, we find that our proposed CARA architecture, which leverages the sequences of users' checkins as well as

[10]The default learning rate setting of the Adam optimiser in Keras.

Table 4: As per Table 3; comparison between our proposed CARA architecture and various factorisation baselines.

	Brightkite		Foursquare		Yelp	
Model	HR	NDCG	HR	NDCG	HR	NDCG
Normal Users						
MF	0.6206*	0.3470*	0.6656*	0.3818*	0.3539*	0.1734*
BPR	0.6890*	0.4333*	0.7550*	0.4834*	0.4992*	0.2691*
GeoBPR	0.7339	0.4672*	0.8216*	0.5395*	0.5570	0.3020*
STELLAR	0.7267*	0.5635*	0.8751*	0.6984*	0.5356*	0.2969*
NueMF	0.7073*	0.5358*	0.8361*	0.5842*	0.4927*	0.2734*
DRCF	0.7363	0.5670*	0.8805	0.6814*	0.5209*	0.2890*
CARA	**0.7385**	**0.6040**	**0.8851**	**0.7154**	**0.5587**	**0.3272**
Cold-Start Users						
MF	0.6768*	0.3913*	0.6623*	0.3650*	0.3748*	0.1868*
BPR	0.7519	0.4907*	0.7792-	0.4961*	0.5273*	0.2946*
GeoBPR	**0.8093**	0.5262*	0.8312	0.5486*	**0.5802**	0.3202*
STELLAR	0.7406*	0.5580*	0.8052-	0.6007-	0.5537*	0.3147*
NueMF	0.7160*	0.5894*	0.7922-	0.6227	0.5102*	0.2956*
DRCF	0.7409*	0.5618*	0.8442	**0.6542**	0.5399*	0.3083*
CARA	0.7648-	**0.6220**	**0.8636**	0.6505	0.5748	**0.3493**

the ordinary and transition contexts associated with the checkins, is effective for CAVR for both normal and cold-start users.

In addressing research questions RQ2 and RQ4, we compare CARA with GRU architectures that consider both the ordinary and transition context (CAGRU, CGRU and LatentCross). Note that these GRU baselines treat the ordinary and transition context similarly and rely on the quantised mapping procedures to represent the contexts. However, as mentioned in Section 3.3.3, we argue that different types of context might influence the user's dynamic preferences differently. In addition, using the mapping procedure to convert the continuous values of the transition context can lead to a loss in granularity. From the results in Table 3, we observe that our proposed CARA architecture that leverages the absolute continuous values of the transition context (i.e. the time interval Δt_τ and the geographical distance Δg_τ – see Section 4.3) is more effective than the CAGRU, CGRU and LatentCross baselines in capturing the transition effects between successive checkins. In particular, our proposed Contextual Attention Gate (CAG) enables the CARA architecture to treat the ordinary and transition separately, while these GRU baselines do not do so.

Next, we compare our proposed CARA architecture with the state-of-the-art factorisation approaches. From the first group of rows in Table 4, we observe that CARA consistently and significantly outperforms all the factorisation baselines across three datasets in terms of HR and NDCG. In particular, comparing with STELLAR, the state-of-the-art CAVR that considers both the contextual information and the sequences of users' checkins, CARA obtains 7.19% and 10.21% improvements in terms of NDCG for Brightkite and Yelp datasets, respectively. In addition, comparing with DRCF, the recent DNN framework that exploits RNN models to capture the users' dynamic preferences, our proposed CARA architecture significantly outperforms DRCF by 6.53%, 5% and 13.22% in terms of NDCG for Brightkite, Foursquare and Yelp datasets, respectively. Furthermore, we also observe that CARA significantly outperforms STELLAR and DRCF by 10-13% in terms of NDCG for the Brightkite and Yelp datasets. We also highlight that GeoBPR uses an advanced geo-based negative sampling technique [32], while CARA uses traditional negative sampling, similar to BPR. CARA is as effective as GeoBPR in terms of HR on Brightkite and Yelp (i.e. no significant differences are observed), while using a less advanced sampling technique. We underline that CARA can be adapted to use GeoBPR's negative sampling, which we leave to future work.

We further investigate the effectiveness of our proposed CARA architecture and the GRU baselines under different settings. In particular, Figure 4 presents the performances on the Brightkite and Yelp datasets[11] – in terms of HR@10 and NDCG@10 – of various GRU architectures, by considering the users with particular time intervals Δt (hours) and geographical distances Δg (km) between their last checkin and ground-truth checkin. Regarding the effectiveness of CARA, we observe that CARA consistently outperforms all GRU baselines in terms of HR and NDCG, while CARA outperforms the GRU baselines in term of NCDG for the Brightkite dataset. In particular, with respect to research question RQ4, CARA consistently outperforms TimeGRU in terms of HR and NDCG across the Brightkite and Yelp datasets. These results imply that our proposed CARA architecture, which consists of Time- and Spatial-based Gates (TSG), is more effective than TimeGRU, the GRU baseline that considers only the time intervals. Therefore, by considering both the time interval and the geographical distance between two successive checkins, CARA can generate better recommendations than TimeGRU. Next, to address research question RQ3, we compare CARA with CGRU and LatentCross, the GRU baselines that rely on the quantised mapping procedures to represent the transition contexts (**Limitation 4**), on the Yelp and Brightkite datasets[11]. The results from Figure 4 demonstrate that our proposed CARA architecture, which supports the continuous values of the transition contexts, outperforms CGRU and LatentCross on both settings (i.e. fixed geographical distances $\Delta g = 1$ km and $\Delta g = 5$ km).

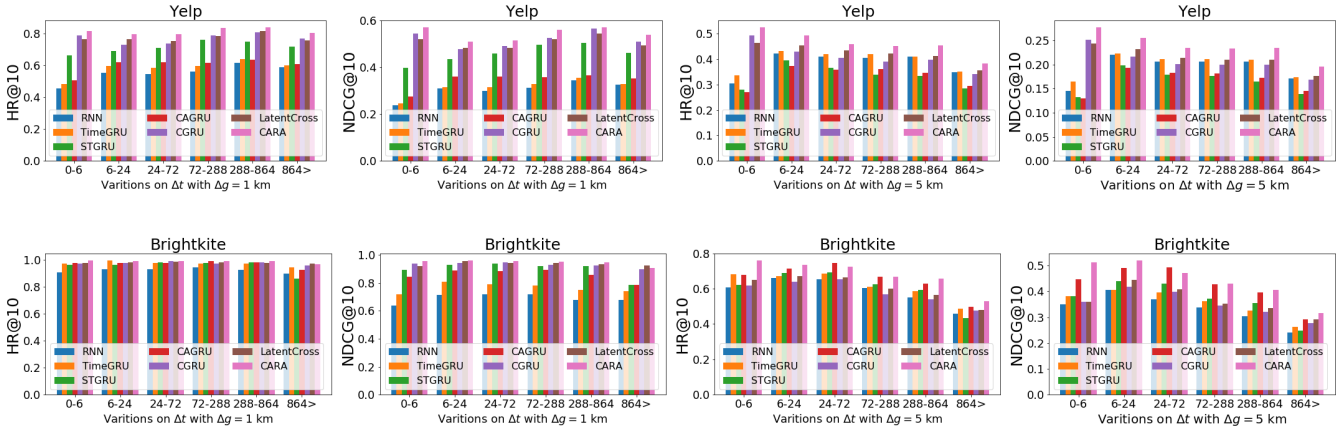
Furthermore, Figure 4 demonstrates that the effectiveness of all approaches for the Brightkite dataset decreases as the time intervals between two successive checkins increases because users are less likely to be influenced by venues they visited long time ago. Moreover, the experimental results using a fixed geographical distance of $\Delta g = 1$ km on both datasets demonstrate the ability of CARA in capturing the users' dynamic preferences (as discussed in Section 1). In particular, even when the time interval between two checkins is long (e.g. more than 864 hours) but the geographical distances are small, CARA still outperforms all baselines, demonstrating the value of learning using nearby checkins as well as recent checkins.

7 CONCLUSIONS

In this paper, we proposed a novel Contextual Attention Recurrent Architecture (CARA) for Context-Aware Venue Recommendation (CAVR), positioned within five elicited limitations with respect to the state-of-the-art GRU architectures that adapt GRU units. In particular, our proposed architecture consists of two gating mechanisms: namely 1) the Contextual Attention Gate (CAG) that controls the influence of the ordinary and transition contexts on the users' dynamic preferences and 2) the Time-and Spatial-based Gates (TSG) that control the influence of the hidden state of previous GRU units based on the time intervals and geographical distances between successive checkins. Our comprehensive experiments on three large-scale datasets from the Brightkite, Foursquare and Yelp commercial LBSNs demonstrate the significant improvements of our proposed CARA architecture for CAVR in comparison with

[11] While Figure 4 only shows results for Brightkite and Yelp, the results for Foursquare – omitted for lack of space – are consistent.

Figure 4: Performance between our proposed CARA architecture and various GRU architectures on the Brightkite and Yelp datasets by varying the time interval Δt in term of hours with the fixed values of the geographical distances Δg (1 and 5 km).



various state-of-the-art GRU architectures, as well as various recent factorisation approaches, in both normal and cold-start settings. Indeed, significantly CARA improves NDCG by 5-13% over the recent DRCF framework [21] across the three datasets. For future work, we plan to extend the CARA architecture to incorporate additional information such as the social relationships between users to further improve the quality of recommendation for CAVR.

REFERENCES

[1] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and H Chi. 2018. Latent Cross: Making Use of Context in Recurrent Recommender Systems. In *Proc. of WSDM.*

[2] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proc. of ICML.*

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, , et al. 2016. Wide & deep learning for recommender systems. In *Proc. of DLRS.*

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proc. of NIPS.*

[5] Bin Cui, Anthony KH Tung, Ce Zhang, and Zhe Zhao. 2010. Multiple feature fusion for social media applications. In *Proc. of SIGMOD.*

[6] Romain Deveaud, M-Dyaa Albakour, Craig Macdonald, and Iadh Ounis. 2014. On the importance of venue-dependent features for learning to rank contextual suggestions. In *Proc. of CIKM.*

[7] Romain Deveaud, M-Dyaa Albakour, Craig Macdonald, and Iadh Ounis. 2015. Experiments with a venue-centric model for personalisedand time-aware venue suggestion. In *Proc. of CIKM.*

[8] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations. In *Proc. of RecSys.*

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. of CVPR.*

[10] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In *Proc. of SIGIR.*

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proc. of WWW.*

[12] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proc. of SIGIR.*

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997).

[14] How Jing and Alexander J Smola. 2017. Neural survival recommender. In *Proc. of WSDM.*

[15] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[16] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009).

[18] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Context-aware sequential recommendation. In *Proc. of ICDM.*

[19] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. In *Proc. of AAAI.*

[20] Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval* (2009).

[21] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2017. A Deep Recurrent Collaborative Filtering Framework for Venue Recommendation. In *Proc. of CIKM.*

[22] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2017. A Personalised Ranking Framework with Multiple Sampling Criteria for Venue Recommendation. In *Proc. of CIKM.*

[23] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. 2016. Phased LSTM: Accelerating recurrent network training for long or event-based sequences. In *Proc. of NIPS.*

[24] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2012).

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. of UAI.*

[26] Elena Smirnova and Flavian Vasile. 2017. Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks. In *Proc. of the DLRS.*

[27] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proc. of DLRS.*

[28] Song Tang, Zhiyong Wu, and Kang Chen. 2017. Movie Recommendation via BLSTM. In *Proc. of ICMM.*

[29] Bartłomiej Twardowski. 2016. Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks. In *Proc. of RecSys.*

[30] Lina Yao, Quan Z Sheng, Yongrui Qin, Xianzhi Wang, Ali Shemshadi, and Qi He. 2015. Context-aware Point-of-Interest Recommendation Using Tensor Factorization with Social Regularization. In *Proc. of SIGIR.*

[31] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proc. of SIGIR.*

[32] Fajie Yuan, Guibing Guo, Joemon Jose, Long Chen, and Haitao Yu. 2016. Joint Geo-Spatial Preference and Pairwise Ranking for Point-of-Interest Recommendation. In *Proc. of ICTAI.*

[33] Hanwang Zhang, Yang Yang, Huanbo Luan, Shuicheng Yang, and Tat-Seng Chua. 2014. Start from scratch: Towards automatically identifying, modeling, and naming visual attributes. In *Proc. of MM.*

[34] Jia-Dong Zhang and Chi-Yin Chow. 2015. GeoSoCa: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *Proc. of SIGIR.*

[35] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks. In *Proc. of AAAI.*

[36] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. 2016. STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation. In *Proc. of AAAI.*

[37] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to Do Next: Modeling User Behaviors by Time-LSTM. In *Proc. of IJCAI.*