Hierarchical Context Enabled Recurrent Neural Network for Recommendation

Kyungwoo Song,[1*] **Mingi Ji,**[1*] **Sungrae Park,**[2] **Il-Chul Moon**[1]
[1]Korea Advanced Institute of Science and Technology (KAIST), Korea
[2]Clova AI Research, NAVER Corp., Korea
{gtshs2,qwertgfdcvb}@kaist.ac.kr, sungrae.park@navercorp.com, icmoon@kaist.ac.kr

Abstract

A long user history inevitably reflects the transitions of personal interests over time. The analyses on the user history require the robust sequential model to anticipate the transitions and the decays of user interests. The user history is often modeled by various RNN structures, but the RNN structures in the recommendation system still suffer from the long-term dependency and the interest drifts. To resolve these challenges, we suggest HCRNN with three hierarchical contexts of the global, the local, and the temporary interests. This structure is designed to withhold the global long-term interest of users, to reflect the local sub-sequence interests, and to attend the temporary interests of each transition. Besides, we propose a hierarchical context-based gate structure to incorporate our *interest drift assumption*. As we suggest a new RNN structure, we support HCRNN with a complementary *bi-channel attention* structure to utilize hierarchical context. We experimented the suggested structure on the sequential recommendation tasks with CiteULike, MovieLens, and LastFM, and our model showed the best performances in the sequential recommendations.

Introduction

A user *history* is a sequence of user orders or clicks, and the history represents the user's interest. Given this user history, many services such as movie recommendations, music streaming services, etc., are interested in recommending the next most likely click item. When we perform this recommendation, it has been assumed that the user's interest can be hierarchically ranging from general interest to a temporary, specific need as shown in Figure 1. Here, these hierarchical interest dynamics are defined as 1) the global context for the entire sequence; 2) the local context for a sub-sequence, such as a click-stream of a site visit with a few or dozens of clicks; 3) and the temporary context for a transition of items. The assumption on the hierarchical contexts has been partially reflected in NARM that models the attention of the general interest (Li et al. 2017); and STAMP that directly predicts the next item by considering temporary contexts (Liu et al. 2018).

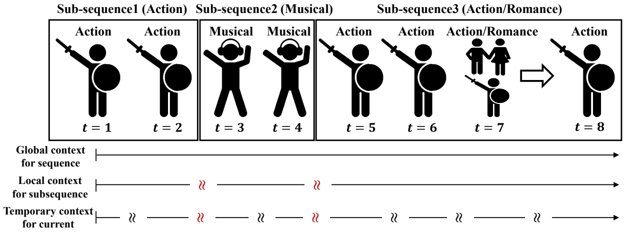


Figure 1: The long user history contains multiple hierarchical context; global context, local context, and temporary context. To take into account the user's interest drift, the temporary context must change at every point (black wave) but should change more when the new sub-sequence starts (red wave). The wave means the change of interest, and red wave means a more drastic change than the black wave. Moreover, the interest drift should be considered in the hierarchical context. For example, in the figure above, we can see that the user's primary interest is the action movie, given the global and local context. Therefore, even if a movie, whose genre is action and romance, comes out at $t = 7$, we can recommend an action movie at $t = 8$ rather than a romance movie if we consider the hierarchical context.

The recently proposed models with recurrent neural network (RNN) structures have focused on modeling the local context of sub-sequences (Wu et al. 2017; Smirnova and Vasile 2017; Yin et al. 2016). For example, GRU4REC (Hidasi et al. 2016) utilized a gated recurrent unit (GRU) (Cho et al. 2014) with a ranking based loss, to emphasize the best item selection. This model started modeling the interest dynamics with general structure, GRU, but the general structure can be further modified to model the hierarchical interest dynamics. Another example is NARM whose attention mechanism is one way of modeling the user's global context. This attention mechanism emphasizes a specific previous record to consider for the next recommendation, and this mechanism enables the long sequence modeling. However, this mechanism could be better if we include modeling on a sudden interest drift of users. In contrast to GRU4REC and NARM, STAMP is optimized to model the short inter-

[*]Equal contribution.

est drift of users without any recurrent structures (Liu et al. 2018). STAMP embeds only the right-before item for temporary interest and the cumulative summary of previous items for general (or global) interest with two feed-forward networks. The STAMP model can be further improved if the structure takes into account the hierarchical interaction between global interest and temporary interest as Figure 1.

This paper proposes Hierarchical Context enabled RNN (HCRNN) which models the hierarchical interest dynamics within a modified RNN structure. To our knowledge, this is the first proposal to operate the hierarchical contexts of interest dynamics with a modified RNN cell structure that optimizes both keeping the global/local context and accepting the temporary drift. HCRNN is similar to the LSTM's mechanism of modeling long-term and short-term memory, separately; but there are inherent differences, as well.

HCRNN does not generate the temporary context from either global or local context. LSTM uses the cell state to produce its corresponding hidden state that is a short-term memory, and with this structure, the hidden state tends to be a subset of cell state. However, if we assume the global interest dynamics can be fundamentally different from the temporary transition, i.e., an sudden purchase order out of consistent long purchase history, we need to separate the long-term memory and the short-term memory.

HCRNN independently maintains the local context and the temporary context, and they interact each other only in the gate (Eq. 15, 17, 18) and attention (Eq. 13) while LSTM does not independently keep the short-term hidden output. For hierarchical context modeling, the global and local contexts need to contain more abstract information than the temporary context. For this purpose, we proposed a new structure to generate the local context that combines the advantages of topic modeling and memory network (Sukhbaatar et al. 2015; Lau, Baldwin, and Cohn 2017).

As shown in Figure 1, it is easy to capture the interest drift of the user with a hierarchical context. In other words, we defined the *interest drift assumption* as "if the user's local context (for sub-sequence) and the current item are very different, the user's temporary interest drift occurs." We proposed a new gate structure to incorporate this assumption effectively. As we propose a modified RNN cell and its outputs with different semantics, we also suggest a modified attention mechanism that is complementary to the proposed cell structure. As the global context becomes a static context, the dynamic context becomes the local and the temporary contexts. Therefore, the attention mechanism will be bi-channel with the local and the temporary contexts, so we named it as the *bi-channel attention*. By the combination of the HCRNN cells, the attention weight with the local context is concentrated on the recent history, and the attention weight with temporary context is distributed to the relatively far history. We have presented an overall structure of HCRNN and bi-channel attention in Figure 2.

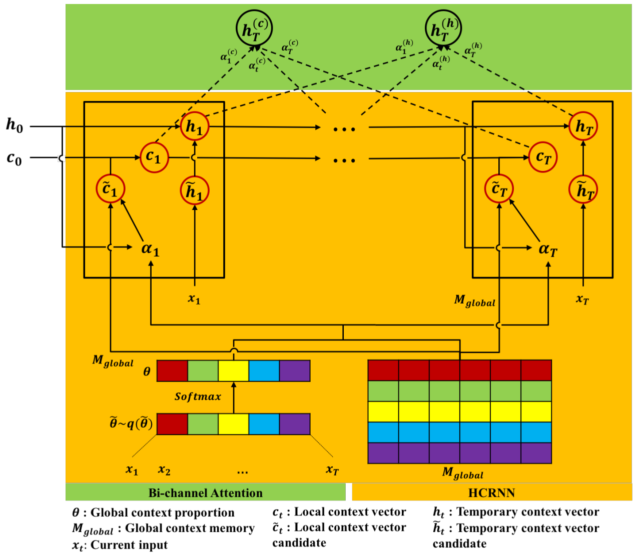


Figure 2: Overall HCRNN and Bi-channel attention structures. Each different color boxes of the θ is the proportion of k-th global context vector, $\theta^{(k)}$. Each row of the M_{global} is the k-th global context vector, $M_{global}^{(k)}$. Same color of box in θ and row in $M_{global}^{(k)}$ mean the same global context.

Preliminary

Cell Structure of Recurrent Neural Networks

LSTM LSTM is a de facto standard of RNNs by enabling the learning from the long-term dependency. A variant of LSTM, or LSTM with peephole connection (Gers, Schraudolph, and Schmidhuber 2002), is a typical LSTM structure with emphasis on the modified gating mechanism by accepting the input from the cell state, and LSTM with peephole are used in previous studies (Zhu et al. 2017; Neil, Pfeiffer, and Liu 2016). The below is the specifications of LSTM with peephole with formulas.

$$i_t = \sigma_i(x_t W_{xi} + h_{t-1} W_{hi} + c_{t-1} \odot w_{ci} + b_i) \tag{1}$$

$$f_t = \sigma_f(x_t W_{xf} + h_{t-1} W_{hf} + c_{t-1} \odot w_{cf} + b_f) \tag{2}$$

$$\widetilde{c}_t = x_t W_{xc} + h_{t-1} W_{hc} + b_c \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \sigma_c(\widetilde{c}_t) \tag{4}$$

$$o_t = \sigma_o(x_t W_{xo} + h_{t-1} W_{ho} + c_t \odot w_{co} + b_o) \tag{5}$$

$$h_t = o_t \odot \sigma_h(c_t) \tag{6}$$

Here, it should be noted that Eq. 6 generates the hidden variable of LSTM, which we consider a temporary context in HCRNN. Eq. 6 does not have any component of h_{t-1} and it means the high dependency of h_t on c_t. This treatment of connection is hard to consider the semantically different context between the local and the temporary context at the same time.

HCRNN modifies the LSTM structure to treat the generation of the local and the temporary context separately to consider the hierarchical contexts at the same time. Besides, we modified the gate structure to consider the interaction between the hierarchical contexts.

GRU GRU is a simplified version of LSTM with fewer parameters while GRU still supports learning from the long-term dependency. GRU replace cell state (c_t) and hidden state (h_t) in LSTM with one hidden state (h_t). The below is the specification of GRU.

$$z_t = \sigma_z(x_t W_{xz} + h_{t-1} W_{hz} + b_z) \tag{7}$$

$$r_t = \sigma_r(x_t W_{xr} + h_{t-1} W_{hr} + b_r) \tag{8}$$

$$\widetilde{h}_t = (r_t \odot h_{t-1}) W_{hh} + x_t W_{xh} + b_h \tag{9}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \sigma_h(\widetilde{h}_t) \tag{10}$$

The enabler of GRU mechanism is Eq. 7 and 8. Hence, when we seek a new gating mechanism to separate the generation of contexts, we were motivated by adopting such condensed gating mechanisms because HCRNN will inevitably increase the number of trained parameters.

Attention on Recurrent Neural Networks

An RNN representing a context up to the present with a fixed length vector suffers from long-term dependency considerations. For that reason, (Bahdanau, Cho, and Bengio 2015) proposed an attention mechanism to retrieve the information needed at present among the past information. The RNN attention mechanism is usually based only on the hidden state of an encoder (h) and decoder (s) of the RNN. $\alpha_{ij} = exp(e_{ij}) / \sum_{k=1}^{T} exp(e_{ik})$ is the attention weight at a point j in time i, and α_{ij} is determined by $e_{ij} = v_a^T \sigma(W_a s_{i-1} + U_a h_j)$. We were motivated by adopting such a hidden state, h, based attention algorithm because consideration of long-term dependency is important for the sequential recommendation. Besides, in recommendation tasks where user interest drifts frequently occur, it is also important to consider the recent user history. For this reason, we have also adopted the local context, c, based attention to account for the recent history in the sub-sequence.

Methodology

This paper introduces HCRNN-1, HCRNN-2, HCRNN-3, and bi-channel attention. First, we will explain the overall structure of HCRNN, followed by a detailed modeling of HCRNN and bi-channel attention in order.

Hierarchical Context Recurrent Neural Network

We propose HCRNN, a modification of the RNN structure, to model three hierarchical contexts optimized for recommendations, which we describe in this section.

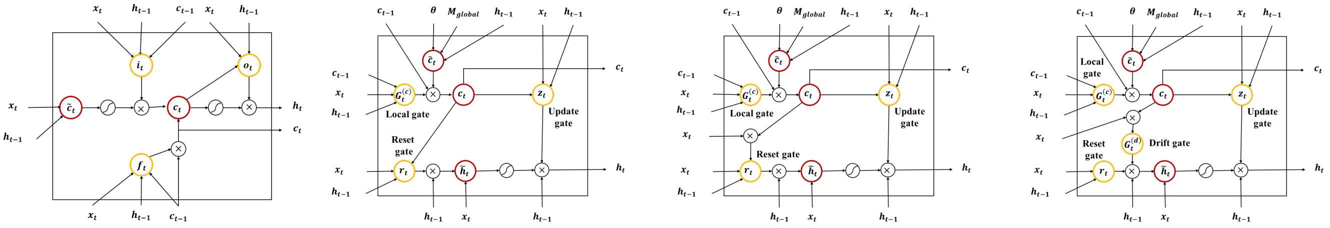
Overall Structure We summarize the overall structure of HCRNN cell at three points. First, h_t of LSTM, which corresponds to the temporary context in HCRNN, is generated by the c_t, which corresponds to the local context in HCRNN. This generation in LSTM indicates that the temporary context is directly influenced by the current cell state, c_t, while HCRNN has no such direct influence to the temporary context, as discussed in *introduction* section. Hence, the generation of the temporary context in HCRNN is detached from the local context, and the local context and temporary context has the connection through the gating mechanism. This makes the creation of temporary contexts more flexible and captures instantaneous interest drift.

Notation	Description
$\|K\|$	Dimension of global context proportion
$\|D\|$	Dimension of item embedding
$\|H\|$	The number of hidden units in HCRNN
$\|I\|$	The number of items
$\|T\|$	The length of the sequence
x_t	The t-th input embedding
$\tilde{c}_t$	The t-th local context candidate
c_t	The t-th local context
$\widetilde{h}_t$	The t-th temporary context candidate
h_t	The t-th temporary context
$G_t^{(c)}$	The t-th local context gate
$G_t^{(d)}$	The t-th drift gate
M_{global}	Global context memory
$M_{global}^{(k)}$	The k-th global context vector
θ	Global context proportion
$\theta^{(k)}$	The k-th global context proportion
r_t	The t-th reset gate of HCRNN
z_t	The t-th update gate of HCRNN
α_t	Global memory attention
$\alpha_t^{(c)}$	Local context attention weight in bi-channel attention
$\alpha_t^{(h)}$	Temoporary context attention weight in bi-channel attention
$W_{c\alpha}^{(1)}, W_{c\alpha}^{(2)}$	Projection matrices for local context attention
$W_{h\alpha}^{(1)}, W_{h\alpha}^{(2)}$	Projection matrices for temporary context attention
W_{emb}	Item embedding matrix
W_B	Weight for bi-linear decoding
$\sigma, \sigma_r, \sigma_z, \sigma_l, \sigma_d$	Sigmoid activation function
σ_h	$\tanh$ activation function

Table 1: The description for the notation in this paper.

Second, we introduce a new static context in the RNN structure as the global context. HCRNN models the global context as two latent variables of M_{global} and θ. Global context memory M_{global} contains global context vector $M_{global}^{(k)}$ for each global context component k. θ is the global context proportion, which means the weight of each global context in the sequence. In other words, $\theta^{(k)}$ is the proportion of activating a certain part of the global context vector, $M_{global}^{(k)}$.

This second modification can be used to generate the local context which contains abstract information. We designed a new unified algorithm for the local context creation for unifying the memory network, the topic modeling, and the recurrent structure with attention. This algorithm has three advantages. 1) The local context is generated from a global context memory,M_{global}, which is a memory network so that it can contain abstract information. 2) The attention used to generate the local context reflects both global context proportion and global context memory. We generate the local



(a) LSTM with peephole (b) HCRNN-1 (c) HCRNN-2 (d) HCRNN-3

Figure 3: HCRNN structure and LSTM with peephole structure. Unlike LSTM, the creation of temporary contexts, h_t, in HCRNN is separated from local contexts, c_t. Also, the and the local contexts, c_t, are designed to be influenced by the item embedding, x_t, only through the gate or attention structure. For this reason, the local context can have a more abstract context than a temporary context. Besides, we propose a new gate structure to incorporate the interest drift assumption relatively strongly to HCRNN-1, HCRNN-2, and HCRNN-3.

context by reflecting the global context proportion at each timestep. For example, if most of the items in a sequence are action movies, the HCRNN is trained to have a high probability in generating a local context associated with the action movie. 3) When the global context is imported into the current local context in Eq. 13,14, we utilize the temporary context, h_t, so a local context, c_t is adapted to the global context influenced by the temporary context. The HCRNN cell in Figure 2 illustrate this hierarchical context generation of M_{global}, θ, c_t, and h_t.

Third, we designed the gating structure to reflect the interest drift assumption by hierarchical contexts. Figure 1 illustrates an occurrence of interest drifts when a user selects an item different from the local context. To reflect this interest drift assumption, we modified the reset gate in HCRNN-2 as shown in Eq. 21. Furthermore, HCRNN-3 has a drift gate, $G_t^{(d)}$ in Eq. 22, with only local context (c_t) and current item embedding (x_t) while the reset gate is influenced by the previous temporary context (h_{t-1}), the local context, and the item embedding, jointly. $G_t^{(d)}$ emphasizes the reset initiated only by the interest drift. Figure 3 shows the overall structure of LSTM and HCRNN-1,2,3 structure.

HCRNN-1 The first version of HCRNN introduces the global contexts and the modified structure from the LSTM cell. First, as we introduced in the previous section, the global context consists of the global context proportion, θ, and global context memory, M_{global}. Here, θ is similar to the topic proportion in general topic models, such as LDA (Blei, Ng, and Jordan 2003), and our model is designed by following the TopicRNN (Dieng et al. 2017). However, unlike TopicRNN, we introduce M_{global} designed as a memory network holding the abstract information, represented as an embedding of each topic, or a global context in the recommendation domain.

The modification consists of two phases. First, to model the local context candidate $\widetilde{c}_t$, we modeled the degree, α_t, to which we should consider for each global context vector, $M_{global}^{(k)}$, at the current time step. α_t is obtained from the attention mechanism, which is different from the bi-channel attention in *Bi-channel Attention and Prediction* section, based on the previous temporary context h_{t-1}, M_{global} and θ in Eq. 13. This attention of α_t is an attention mechanism within the HCRNN cell, yet the bi-channel attention is attention outside of the HCRNN cell sequence. Because α_t is computed with the temporary context of h_{t-1}, the local context candidate, $\widetilde{c}_t$, can fluctuate temporarily. To handle this fluctuation, we formulate a local gate, $G_t^{(c)}$ in Eq. 15, 16. The local gate $G_t^{(c)}$, helps local context to change more stable, different with temporary context.

The second phase is modeling the temporary context, h_t, with the current input, x_t, and the previous temporary context, h_{t-1}. The temporary context does not directly come from the local context of c_t and the global context of θ, M_{global}. However, the reset gate of r_t uses the local and the temporary contexts to reset the components of the temporary context. Additionally, the update gate of z_t controls the update with the current input, the local and the temporary contexts. This structure allows the temporary context to focus more on the current input, unlike the local context.

$$\widetilde{\theta} \sim q(\widetilde{\theta}) = \mathcal{N}(\widetilde{\theta}; \mu(x_{1:T}), \mathrm{diag}(\sigma^2(x_{1:T}))) \tag{11}$$

$$\theta \sim \mathrm{softmax}(\widetilde{\theta}) \tag{12}$$

$$\alpha_t^{(k)} = \mathrm{softmax}(v_\theta^T \sigma(h_{t-1}W_{h\alpha} + (\theta^{(k)} M_{global}^{(k)})W_{\theta\alpha})) \tag{13}$$

$$\widetilde{c}_t = \sum_{k=1}^{K} \alpha_t^{(k)} M_{global}^{(k)} \tag{14}$$

$$G_t^{(c)} = \sigma_l(x_t W_{xl} + h_{t-1} W_{hl} + c_{t-1} W_{cl} + b_l) \tag{15}$$

$$c_t = (1 - G_t^{(c)}) \odot c_{t-1} + G_t^{(c)} \odot \widetilde{c}_t \tag{16}$$

$$z_t = \sigma_z(x_t W_{xz} + h_{t-1} W_{hz} + c_t W_{cz} + b_z) \tag{17}$$

$$r_t = \sigma_r(x_t W_{xr} + h_{t-1} W_{hr} + c_t W_{cr} + b_r) \tag{18}$$

$$\widetilde{h}_t = (r_t \odot h_{t-1}) W_{hh} + x_t W_{xh} + b_h \tag{19}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \sigma_h(\widetilde{h}_t) \tag{20}$$

μ and σ^2 for the normal distribution denote the output of inference network as defined in Eq. 32, 33. θ denotes the global context proportion and its dimension is $|K|$. M_{global} denotes the global context memory with $|K| \times |D|$ size. x_t and c_t denote the current item embedding and the local context vector respectively and they are $|D|$-dimensinoal vector. h_t denote the temporary context vector with dimension $|H|$.

Table 1 denotes the notation for HCRNN.

HCRNN-2 After we suggest HCRNN-1, we update the generation of temporary contexts, h_t, by modifying the reset gate with the local context, c_t; the temporary context, h_t; and the input, x_t. Under the interest drift assumption, the interest drift can be identified if the local context and the current input are very different. If the interest drift occurred, we need to further update the temporary context, h_t, by reducing the information from h_{t-1}, than the case without the drift. Thus, the comparison between the local context and the current input is necessary to gauge the necessity of h_{t-1}. Finally, we substitute Eq. 18 with Eq. 21 as the below.

$$r_t = \sigma_r(x_t W_{xr} + h_{t-1} W_{hr} + (x_t \odot c_t) W_d + b_r) \\ s.t. W_d \geq 0 \tag{21}$$

Eq. 21 design the reset gate, so that the r_t becomes small as the element-wise product between c_t and x_t decreases because of interest drift. This similarity magnitude needs to be scaled and regularized, so we multiply a constraint $W_d \geq 0$. Also, we used a projection operator (Rakhlin, Shamir, and Sridharan 2012) to handle the constraint.

HCRNN-3 The reset gate in HCRNN-2 reflects the interest drift assumption on updating the temporary context. This update in HCRNN-2 requires x_t, h_{t-1}, and c_t to be mixed to generate the signal of the reset gate. The suggested update linearly models the relevance between the local context, c_t, and the current input, x_t. However, the linear activation from the element-wise product may not embody the binary nature of the temporary drift. Hence, we add a sigmoid activation on top of the element-wise product, which eventually becomes an independent gate, $G_t^{(d)}$, to model the interest drift.

Since the sigmoid function outputs a value between 0 and 1, the reset gate of HCRNN-2 in Eq. 21 can have a value between 0 and 1 theoretically. However, the sigmoid function is not sharp, and it makes the most LSTM forget gate values (similar to reset gate in GRU) are experimentally located in the middle state (0.5) (Li et al. 2018). In fact, in our experiments, the reset gate, r_t in Eq. 21 is on the average 0.47 (± 0.03) on the CiteULike dataset. To make the temporary context focus on x_t, the value of gate multiplied by h_{t-1} in Eq. 19 need to be smaller. We model the new interest drift gate (Eq. 22) and use the product of Eq. 22 and 23 in Eq. 24. The product of Eq. 22 and 23 has a value of 0.29 (± 0.021) on average, and it is 38.2% smaller than that of Eq. 21.

$$G_t^{(d)} = \sigma_d((x_t \odot c_t) W_d + b_d) \quad s.t. W_d \geq 0 \tag{22}$$

$$r_t = \sigma_r(x_t W_{xr} + h_{t-1} W_{hr} + b_r) \tag{23}$$

$$\widetilde{h}_t = (r_t \odot (G_t^{(d)} \odot h_{t-1})) W_{hh} + x_t W_{xh} + b_h \tag{24}$$

Eq. 22, 23 lets the temporary context be more affected by the current input when the temporary drift is captured by the drift gate, $G_t^{(d)}$.

Bi-Channel Attention and Prediction

As mentioned in the *Introduction*, it is important to learn from both long-term dependency and recent interest in a sequential recommendation. One common technique to emphasize the long-term dependency is an attention mechanism, but we introduce modified attention given a HCRNN cell structure because of its hierarchical contexts. To exploit the hierarchical contexts of HCRNN, we implement the complementary bi-channel attention as the local context attention, $\alpha_t^{(c)}$; and the temporary context attention $\alpha_t^{(h)}$.

Both $\alpha_t^{(c)}$ and $\alpha_t^{(h)}$ needs to result in a higher attention weight if two compared context vectors are similar. $\alpha_t^{(h)}$ is modeled as a conventional linear sum based alignment function, so the training on the weight parameter can select which to attend in the temporary context. The projection matrices for $\alpha_t^{(h)}$ are $W_{h\alpha}^{(1)}, W_{h\alpha}^{(2)}$ which are both $|H| \times |H|$ matrix. Besides, we implemented the scaled dot-product based attention function (Vaswani et al. 2017) for $\alpha_t^{(c)}$ because the dot-product will maximize the attention with the same local context vectors. This modeling will produce a stronger attention weight to the items in the similar sub-sequence. The projection matrices for $\alpha_t^{(c)}$ are $W_{c\alpha}^{(1)}, W_{c\alpha}^{(2)}$ which are both $|D| \times |H|$ matrix.

We used a concatenation of h_t, $h_t^{(c)}$, and $h_t^{(h)}$ for the appropriate item prediction with a bi-linear decoding scheme following NARM as in Eq. 28. W_{emb} is an item embedding, and W_B is a weight for bi-linear decoding. We calculated the prediction-related loss through cross-entropy.

$$\alpha_{tj}^{(c)} = \text{softmax}\left(\frac{(c_t W_{c\alpha}^{(1)})(c_j W_{c\alpha}^{(2)})^T}{\sqrt{|H|}}\right) \tag{25}$$

$$\alpha_{tj}^{(h)} = \text{softmax}(v_h^T \sigma(h_t W_{h\alpha}^{(1)} + h_j W_{h\alpha}^{(2)})) \tag{26}$$

$$h_t^{(c)} = \sum_j \alpha_{tj}^{(c)} h_j \quad \text{and} \quad h_t^{(h)} = \sum_j \alpha_{tj}^{(h)} h_j \tag{27}$$

$$\widehat{y}_t = \text{softmax}(W_{emb}^T W_B [h_t, h_t^{(c)}, h_t^{(h)}]) \tag{28}$$

Model Inference

While training the local and the temporary contexts relies on the gradient method with a deterministic learning, HCRNN includes the global context which follows the topic probabilistic model, such as LDA, VAE, and GSM (Kingma and Welling 2014; Miao, Grefenstette, and Blunsom 2017). This generative modeling requires a maximization on the log-marginal likelihood of Eq. 29, so we utilize the variational inference by optimizing the evidence lower bound (ELBO) of Eq. 30 (Jordan et al. 1999).

$$\log p(y_{1:T}|c_{1:T}, h_{1:T}) = \log \int p(\widetilde{\theta}) \prod_{t=1} p(y_t|\widetilde{\theta}, c_t, h_t) d\widetilde{\theta} \tag{29}$$

$$\geq \sum_{t=1}^{T} E_{q(\widetilde{\theta})}[\log p(y_t|\widetilde{\theta}, c_t, h_t)] - \text{KL}[(q(\widetilde{\theta})||p(\widetilde{\theta}))] \tag{30}$$

The variational inference of HCRNN assumes the variational distribution, q, that is a feed-forward neural network. Following the VAE framework, q also becomes the amortized inference network with the input, $x_{1:T}$, to predict μ and $\log \sigma$. Specifically, the prediction of μ is done by Eq. 32, and $\log \sigma$ by Eq. 33, where f is a feed-forward neural network.

$$q(\widetilde{\theta}) = \mathcal{N}(z; \mu(x_{1:T}), \mathrm{diag}(\sigma^2(x_{1:T}))) \tag{31}$$

$$\mu(x_{1:T}) = W_q^{(1)} f(x_{1:T}) + b_q^{(1)} \tag{32}$$

$$\log \sigma(x_{1:T}) = W_q^{(2)} f(x_{1:T}) + b_q^{(2)} \tag{33}$$

After the inference on μ and $\log \sigma$, the sampled $\widetilde{\theta}$ is used as the global context after turning it into θ by the softmax function. Our HCRNN source code is available at https://github.com/gtshs2/HCRNN.

Experimental Result

Datasets For the performance evaluation, we used three publicly available datasets: CiteULike, LastFM, and MovieLens[1]. We aim at modeling a long user history, so we removed sequences whose length is less than 10. Besides, we removed the items that exist only in the test set, and the items that appeared less than 50/50/25 times in three datasets respectively. We performed cross-validation by assigning 10% of the randomly chosen train set as the validation set. We also followed the data augmentation method as proposed in NARM (Li et al. 2017) and improved GRU4REC (Tan, Xu, and Liu 2016). The data augmentation techniques can enhance the performance by reducing the overfitting. Table 2 summarizes the descriptive statistics of preprocessed datasets.

Dataset	CiteULike	LastFM	MovieLens
# sequence(train)	38,724	73,420	136,233
# sequence(test)	9,140	17,829	34,682
# clicks	1,163,813	4,575,159	5,041,882
# items	1,980	5,778	930
avg. len	24.31	50.14	29.50

Table 2: Statistics of evaluation datasets.

Baselines We compared HCRNN with the below eight baselines.

- **POP** exploits the frequency of items in the training set. It always recommends items that appear most often in the training set.
- **SPOP** is Similar to POP, S-POP also exploits the frequency, but it recommends items that appear most often in the current sequence.
- **Item-KNN** (Davidson et al. 2010; Linden, Smith, and York 2003) recommends items based on the co-occurrence number of item pairs, and Item-KNN interprets the co-occurrence as a similarity. The model recommends similar items only in the same sequence.
- **BPR-MF** (Rendle et al. 2009) is a model representing a group of models with matrix factorization (MF) and Bayesian personalized ranking loss (BPR). By introducing the ranking loss, BPR-MF shows a better performance than a typical MF in the recommendation.
- **GRU4REC** (Hidasi et al. 2016) is a sequential model with GRUs for the recommendation. This model adopts a session parallel batch and a loss function such as Cross-Entropy, TOP1, and BPR.
- **LSTM4REC** is our version of a GRU4REC variant with LSTM.
- **NARM** (Li et al. 2017) is a model based on GRU4REC with an attention to consider the long-term dependency. Besides, it adopts an efficient bi-linear loss function to improve the performance with fewer parameters.
- **STAMP** (Liu et al. 2018) considers both current interest and general interest of users. In particular, STAMP used an additional neural network for the current input only to model the user's current interest. Also, it proposes a tri-linear loss function.

Experiment Settings For fair performance comparisons, we set the batch size (512), the item embedding (100), the RNN hidden dimension (100), the input dropout (0.25), the output layer dropout (0.5), the optimizer (Adam), and the learning rate (0.001)[2] as shown in NARM (Li et al. 2017).

Quantitative Performance Evaluation

Table 3 shows the performance of the baselines and HCRNN with two measurements of *recall* at K (R@K) and *mean reciprocal ranking* at K (M@K), which are widely used in the sequential recommendation. We varied K by 3 and 20. The experiments on HCRNN has an ablation study variation of HCRNN-1, HCRNN-2, HCRNN-3, and HCRNN-3 with bi-channel attentions (HCRNN-3+Bi). The quantitative evaluation indicates that the variations of HCRNN have significant performance improvements in all data and metrics. Particularly, HCRNN-3 with the bi-channel attentions always exhibits the best performance. Additionally, the better performance of HCRNN over NARM, which also has a context modeling, may suggest the need for hierarchical context modeling in recommendations. Moreover, HCRNN shows the best result compared to the RNN based recommendations, i.e., NARM, GRU4REC, and LSTM4REC, so the modified HCRNN cell may have contributed to the performance improvements. As HCRNN-3 with drift gate, $G_t^{(d)}$, shows better results than HCRNN-1 and HCRNN-2, our interest drift assumption may be experimentally justifiable. As HCRNN-3+Bi is the best case, we justify that bi-channel attention with hierarchical contexts may improve the performance experimentally. Finally, NARM with RNN and attention shows better performance than STAMP with a feed-forward neural network. This demonstrates the importance of sequential modeling in recommendations with long sequences.

Qualitative Analysis

From the sensitivity perspective, the global context is unlikely to change given a single item. The local context should

[1] We converted it into a binary implicit rating by activating only the maximum rating.

[2] For STAMP, we set it to 0.005 as shown in the STAMP paper.

	CiteULike				LastFM				MovieLens			
	R@3	R@20	M@3	M@20	R@3	R@20	M@3	M@20	R@3	R@20	M@3	M@20
POP	1.44	5.78	0.92	1.44	0.37	1.99	0.34	0.51	2.43	12.51	1.54	2.65
S-POP	1.26	4.99	0.79	1.23	0.87	3.65	0.55	0.87	2.27	12.23	1.42	2.52
Item-KNN	0.00	6.90	0.00	4.79	0.00	11.59	0.00	8.00	0.00	6.32	0.00	4.28
BPR-MF	0.49	3.15	0.27	0.60	0.82	2.15	0.59	0.73	1.69	8.93	1.07	1.91
LSTM4REC	7.07	23.33	4.93	6.82	15.29	24.75	12.68	13.95	8.52	32.80	5.63	8.45
GRU4REC	8.37	24.19	5.98	7.86	18.29	26.46	15.85	16.95	8.50	32.74	5.60	8.42
NARM	7.81	24.82	5.40	7.41	18.30	33.60	13.12	15.25	9.14	33.42	6.09	8.93
STAMP	5.09	21.93	3.25	5.22	9.29	19.84	6.62	8.01	3.95	20.52	2.65	4.47
HCRNN- 1	8.60	25.36	6.18	8.16	20.67*	34.40*	15.77	17.68*	9.23	33.78*	6.13	9.00
HCRNN- 2	8.83	25.10	6.41*	8.38*	20.78*	34.14*	16.20	18.08*	9.22	33.76*	6.14	9.01
HCRNN- 3	9.21*	25.42*	6.65*	8.61*	21.39*	34.72*	16.66*	18.52*	9.38*	33.67*	6.23*	9.08*
HCRNN-3 + Bi	**9.33***	**25.81***	**6.74***	**8.70***	**21.90***	**34.80***	**17.33***	**19.12***	**9.53***	**33.83***	**6.38***	**9.21***
Improvement(%)	11.47	3.99	12.71	10.69	19.67	3.57	9.34	12.80	4.27	1.23	4.76	3.14

Table 3: Performance evaluation of the proposed models. The boldface indicates the best result among our models and the underline indicates the best result among the baselines. $P^* < 0.05$ (Student's t-test)

change when the item selection is dissimilar to the previous selection, but if the selections are similar, the local context does not change much by Eq. 13-16. The temporary context likely changes for each selected item to represent the current interest and the temporary context significantly changes when the genre transition happens. Because of the two gating structure of Eq. 24 modeling, the average amount of change in the temporary context is more significant than that of the local context as our assumption and expectation. Experimentally, with CiteULike, the temporary context, h, changes 0.278(± 0.037) on average at every timestep, and the local context, c, changes by 0.005(± 0.024) on average.

Context Embedding This study models the hierarchical context, global context memory (M_{global}), local context (c_t), and the temporary context (h_t). The local context is generated by the global context memory, and the temporary context is generated by the previous temporary context and the current item embedding (x_t). The first analysis is visualizing the global context memory (M_{global}) and the item embedding (x_t), to verify the quality of inputs to the construction on the local context (c_t) and the temporary context (h_t). Figure 4 is the joint visualization of the item and the global context memory. The item embeddings are coherently organized as a cohesive cluster with the same genre, and the global context memory covers most of the area that the item embeddings are dispersed. Given the item and the global context memory, we calculated the cosine similarity, and Table 4 enumerates the most aligned items with a specific global context vector in the global context memory.

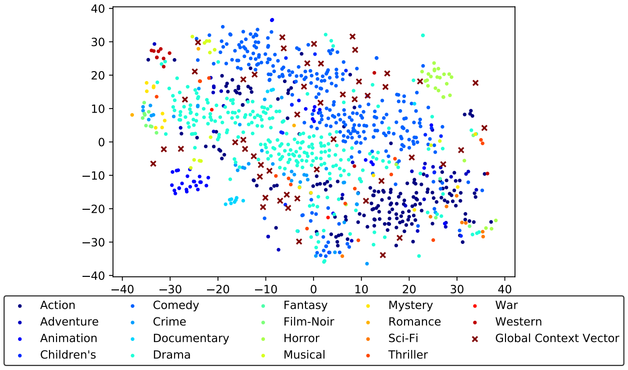


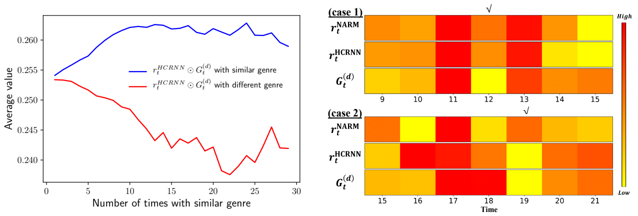
Figure 4: Item embedding and global context vector, $M^{(k)}_{global}$, visualization with tSNE(van der Maaten and Hinton 2008). Item embedding is interpretable with genre, and global context vector cover the most of the items.

Gate Analysis HCRNN is a model to capture the user's interest drift with hierarchical context and drift gate $G^{(d)}_t$. For the comparison of HCRNN and NARM gate structures, we define r^{HCRNN}_t and r^{NARM}_t as the HCRNN and NARM reset gates, respectively. In order to incorporate the interest drift assumption, we designed the value of $r^{HCRNN}_t \odot G^{(d)}_t$ gate applied to h_{t-1} to be smaller when updating $\widetilde{h}_t$ in Eq. 24 if interest drift occurred. Figure 5 represents the gate value when the user clicks the same genre of an item as the previous step and when the user does not. The x-axis in Figure 5a represents the number of consecutive items which has the same genre until the right before timestep. In general, if the genre of the current input is different with previous items, $r^{HCRNN}_t \odot G^{(d)}_t$ has a smaller value compared to the opposite situation. Besides, when a user clicked the same genre of items consecutively, and instantly clicks a different genre of items, the value of $r^{HCRNN}_t \odot G^{(d)}_t$ becomes smaller.

Bi-Channel Attention As mentioned in section *Introduction*, it is important to understand both long-term dependency and recent interests in recommendations with a long user history. Therefore, we present the bi-channel attentions from local and temporary contexts *Bi-Channel Attention and Prediction*, and we present the result in Figure 6. Figure 6a shows the averaged attention weights over the test user histo-

	Genre	Movie Title
$M^{(6)}_{global}$	Animation	Pinocchio, Yellow Submarine, Snow White and the Seven Dwarfs
$M^{(19)}_{global}$	Action	Star Trek: Generations, Predator, Butch Cassidy and the Sundance Kid
$M^{(31)}_{global}$	Horror	Scream, An American Werewolf in London, Dracula

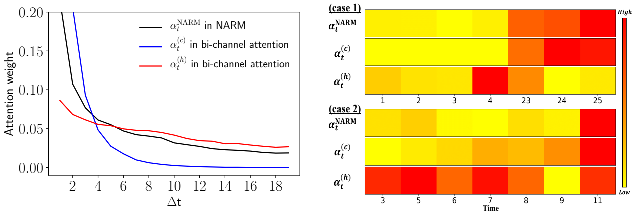
Table 4: Interpretation of global context vector. We listed the items (movie title) close to each global context vector.



(a) Average value of $r_t^{HCRNN} \odot G_t^{(d)}$ gate after appearing items with similar genre consecutively.

(b) Gate heatmap for a user history as an example. We mark "check" when the genre of item changes.

Figure 5: HCRNN takes a large value of $r_t^{HCRNN} \odot G_t^{(d)}$ if the current input of item has similar genre with previous input of item. In the opposite case, HCRNN grasps the user's interest drift and changes $r_t^{HCRNN} \odot G_t^{(d)}$ to smaller.



(a) Averaged attention weight over time difference. Δt means a time difference between prediction time step and the timestep of the previous user history.

(b) Attention heatmap for a user history. The first row of each case is an attention weight in NARM, and the two below are our bi-channel attention weights.

Figure 6: Temporary context based attention, $\alpha_t^{(h)}$, in HCRNN is spread over a long period relatively. Local context based attention, $\alpha_t^{(c)}$, in HCRNN has a large value on recent user records.

ries. NARM has a single attention mechanism, so NARM attention weight, α_t^{NARM}, cannot differentiate the attentions on the local and the temporary contexts. However, the bi-channel attentions distinguishes the attentions for the sub-sequence continuation, $\alpha_t^{(c)}$; and for the temporary transitions, $\alpha_t^{(h)}$. Figure 6a and 6b indicates that $\alpha_t^{(c)}$ focuses on the neighbor attention to check the continuation; and that $\alpha_t^{(h)}$ spreads out through the whole sequence to check the similar temporary transition.

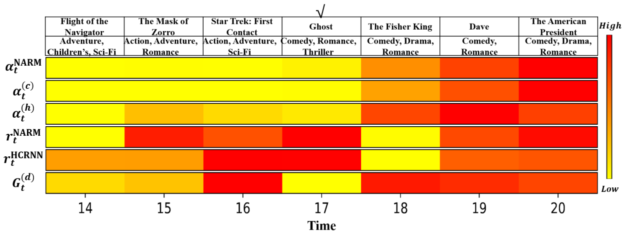


Figure 7: Attention, gate value in NARM and HCRNN, and the change of context value in HCRNN overtime. The drift gate $G^{(d)}_{t=17}$ in HCRNN captured the temporary interest drift

Case Study Figure 7 shows the attention weights, and the gate values for selected user history. In Figure 7, the top three rows represent the attention weights comparing NARM and HCRNN. The bi-channel attentions of HCRNN results in two rows of attentions. The attention of local contexts, $\alpha_t^{(c)}$, focuses on recent history, and the attention of temporary contexts, $\alpha_t^{(h)}$, considers relatively far history. These mean that $\alpha_t^{(c)}$ emphasizes belonging to the same sub-sequence, and $\alpha_t^{(h)}$ tries to find the similar transition throughout the entire history. In particular, $\alpha^{(h)}_{t=15}$ has a relatively high attention weight compared to $\alpha^{(c)}_{t=15}$ and $\alpha^{NARM}_{t=15}$. The rational behind this temporary high attention originates from the same genre of the item entered as input at the last timestep, $t = 20$, whose genre is Romance.

After we observe the attention weights, we observe the gate values, which are r_t^{NARM}, $G_t^{(d)}$, and r_t^{HCRNN}, to verify that the gate operates as we expected. We observed that $G^{(d)}_{t=17}$ has a relatively small value. This small value is caused by the selection of items disaligned to the previous sub-sequence at $t = 16$. This phenomenon demonstrates that $G_t^{(d)}$ can capture the interest drift to reset the temporary context to accept further information from the current item embedding of x_t, as designed in Eq. 22. As $G_t^{(d)}$ is controlled by the local context, the discontinuation of genre matters in $G_t^{(d)}$. However, r_t^{HCRNN}, which also controls the reset of the temporary context in Eq. 23, is not activated because r_t^{HCRNN} only takes the temporary context as the inputs, so the discontinuation does not matter in r_t^{HCRNN}. This rationale applies to r_t^{NARM}, as well. On the contrast, the user history has the same genre at $t = 18, 19, 20$, so $G_t^{(d)}$ also keeps high gate values to prevent the reset on the temporary context.

Conclusion

This paper proposes HCRNN to model the hierarchical contexts for recommendations. We have separated the creation of temporary contexts and local contexts, and it helps the temporary context to focus on the more current item and transient interest. For effective hierarchical context modeling, we present a new context generation structure that utilizes the advantages of the latent topic model and the memory network to contain the abstract information for the global and the local contexts. We also propose the new gate mechanism to incorporate the interest drift assumption. To support HCRNN with hierarchical contexts, we propose bi-channel attentions to account for both long-term dependency and recent interest in the long user history.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1C1B6008652).

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Davidson, J.; Liebald, B.; Liu, J.; Nandy, P.; Van Vleet, T.; Gargi, U.; Gupta, S.; He, Y.; Lambert, M.; and Livingston, B. 2010. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, 293–296. ACM.

Dieng, A. B.; Wang, C.; Gao, J.; and Paisley, J. 2017. Topicrnn: A recurrent neural network with long-range semantic dependency. *International Conference on Learning Representations*.

Gers, F. A.; Schraudolph, N. N.; and Schmidhuber, J. 2002. Learning precise timing with LSTM recurrent networks. *Journal of machine learning research* 3(Aug):115–143.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016. Session-based recommendations with recurrent neural networks. *International Conference on Learning Representations*.

Jordan, M. I.; Ghahramani, Z.; Jaakkola, T. S.; and Saul, L. K. 1999. An introduction to variational methods for graphical models. *Machine learning* 37(2):183–233.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. *International Conference on Learning Representations*.

Lau, J. H.; Baldwin, T.; and Cohn, T. 2017. Topically driven neural language model. *arXiv preprint arXiv:1704.08012*.

Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1419–1428. ACM.

Li, Z.; He, D.; Tian, F.; Chen, W.; Qin, T.; Wang, L.; and Liu, T.-Y. 2018. Towards binary-valued gates for robust lstm training. *arXiv preprint arXiv:1806.02988*.

Linden, G.; Smith, B.; and York, J. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* (1):76–80.

Liu, Q.; Zeng, Y.; Mokhosi, R.; and Zhang, H. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1831–1839. ACM.

Miao, Y.; Grefenstette, E.; and Blunsom, P. 2017. Discovering discrete latent topics with neural variational inference. *International Conference on Machine Learning*.

Neil, D.; Pfeiffer, M.; and Liu, S.-C. 2016. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*, 3882–3890.

Rakhlin, A.; Shamir, O.; and Sridharan, K. 2012. Making gradient descent optimal for strongly convex stochastic optimization. *International Conference on Machine Learning*.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, 452–461. AUAI Press.

Smirnova, E., and Vasile, F. 2017. Contextual sequence modeling for recommendation with recurrent neural networks. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, 2–9. ACM.

Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, 2440–2448.

Tan, Y. K.; Xu, X.; and Liu, Y. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 17–22. ACM.

van der Maaten, L., and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(Nov):2579–2605.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wu, C.-Y.; Ahmed, A.; Beutel, A.; Smola, A. J.; and Jing, H. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, 495–503. ACM.

Yin, H.; Zhou, X.; Cui, B.; Wang, H.; Zheng, K.; and Nguyen, Q. V. H. 2016. Adapting to user interest drift for poi recommendation. *IEEE Transactions on Knowledge and Data Engineering* 28(10):2566–2581.

Zhu, Y.; Li, H.; Liao, Y.; Wang, B.; Guan, Z.; Liu, H.; and Cai, D. 2017. What to do next: Modeling user behaviors by time-lstm. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 3602–3608.