Joint Modeling Dynamic Preferences of Users and Items Using Reviews for Sequential Recommendation

Tianqi Shang[1], Xinxin Li[2], Xiaoyu Shi[3](✉), and Qingxian Wang[2]

[1] College of Computer Science, Sichuan University, Chengdu 610207, China
[2] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
201921090312@std.uestc.edu.cn, qxwang@uestc.edu.cn
[3] Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China
xiaoyushi@cigit.ac.cn

Abstract. The emerging of sequential recommender (SR) has attracted increasing attention in recent years, which focuses on understanding and modeling the temporal dynamic of user behaviors hidden in the sequence of user-item interactions. However, with the tremendous increase of users and items, SR still faces several challenges: (1) the hardness of modeling user interests from spare explicit feedback; (2) the time and semantic irregularities hidden in the user's successive actions. In this study, we present a neural network-based sequential recommender model to learn the temporal-aware user preferences and item popularity jointly from reviews. The proposed model consists of the semantic extracting layer and the dynamic feature learning layer, besides the embedding layer and the output layer. To alleviate the data sparse issue, the semantic extracting layer focuses on exploiting the enriched semantic information hidden in reviews. To address the time and semantic irregularities hidden in user behaviors, the dynamic feature learning layer leverages convolutional fitters with varying size, integrating with a time-ware controller to capture the temporal dynamic of user and item features from multiple temporal dimensions. The experimental results demonstrate that our proposed model outperforms several state-of-art methods consistently.

Keywords: Sequential recommendation · Preference modeling · Temporal dynamic · Semantic extracting · Deep learning

1 Introduction

In the era of information explosion, recommender systems (RS) are the essential enabler for online services and widely applied in a variety of fields, e.g., music/video recommendation, news push-delivery, online shopping. Up to now,

K. Karlapalem et al. (Eds.): PAKDD 2021, LNAI 12713, pp. 524–536, 2021.
https://doi.org/10.1007/978-3-030-75765-6_42

RS can be divided into two categories: general recommender system and sequential recommender system. The general recommender systems with representation of the content-based and collaborative filter (CF)-based solutions seek to capture users' long-term preference, presuming that the user-item interactions in a static way or change slowly over time. Among them, Factorization-based CF methods [5] are the most popular techniques in this era. However, the user preferences are drifting over time, rather than fixed. Different with the general recommender systems heavily focus on modeling users' long-term preference, sequential recommender system can capture the users' short-term preference for more accurate recommendation, with consideration of the sequential dependencies in the user-item interactions [6,11,16]. Hence, SR receives considerable attention due to its superiority in capturing item-to-item sequential relations.

Following this line, several solutions based on neural networks have been proposed to learn the users' short-term preferences. They can be divided into convolutional neural network (CNN)-based and recurrent neural network (RNN)-based methods. CNN-based methods [12] utilize the convolutional filters with different kernel sizes to learn the short-term contexts for recommendation. RNN-based methods include long short-term memory network (LSTM) [15,17] and gated recurrent network (GRU) [2]. They capture the short-term user preference via hidden state of RNN.

Although the previous solutions have achieved satisfactory results, there are still some challenges for sequential recommenders: (1) *the hardness of modeling the user interests from spare explicit feedback*. In most past studies, ratings are used as the only criterion of feedback information to measure the degree of user preference for the specified item. User reviews as another source of data usually contain more information than ratings. For example, "*I really like the style of this skirt and its length is just right. Although the white one is out of stock, I am also satisfied to buy the pink.*" Through the above review, we find out the customer is satisfied with the style rather than the color. Meanwhile, we can also infer that the user is interested in white. However, the rating of "5" cannot show such plentiful information. Therefore, a review-based recommender system can capture more information about user preferences. (2) *the time and semantic irregularities are hidden in the user's successive actions*. The existing temporal-aware recommenders [6,16] always assume that the items in sequence can be considered as evenly spaced and semantically consistent when designing recommender. In practical, the sequence of user behavior is complex, the time intervals between two adjacent interactions can be various. For example, the historical interaction sequence of a user is: $H = \{(i_1, Apr\ 1st), (i_2, Apr\ 2nd), (i_3, May\ 2nd)\}$. It is more reasonable that the information hidden between (i_1, i_2) more than (i_2, i_3), because the user behaviors happen on (i_1, i_2) is just one day, while for the (i_2, i_3) is one month. Furthermore, the sequence of user behavior on items may not share the same semantic topic. For example, the items set of a user interaction is $\{iPhone, iPad, umbrella, dress\}$, the first two items indicate that the user is happy to buy electronic products, while the second two items have no such signal.

To track the above challenges, we present a novel neural network-based sequential recommendation model to learn both long- and short-term user preferences and item popularity jointly from reviews. To evaluate the performance of our proposed model, we conducted a large number of experiments on three datasets from Amazon. Experimental results show that our model is significantly better than the state-of-the-art methods. In summary, our contributions in this paper are as follows:

– We propose a neural network-based model sequential recommender to model the temporal dynamic of user preference and item popularity, which can learn the enrich semantic and temporal information hidden in users' reviews jointly;
– We introduce the dynamic feature learning layer with integrate of time-aware controller to solve the time and semantic irregularities hidden in the user behaviors. It leverages a couple of convolutional fitters with varying sizes to effectively learn the user and item features from multiple temporal dimensions;
– We implement our proposed model on the three datasets from Amazon, and experimental results show that our model is significantly better than state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 summaries the related work on related recommender methods. Section 3 describes the proposed neural network-based model in detail. The experimental setting and results are given in Sect. 4. In Sect. 5, we conclude our paper.

2 Related Work

2.1 Review-Based Recommender

User reviews, can potentially alleviate the data sparsity problem caused by rating-based methods. Bao et al. [1] proposed a novel matrix factorization model (called TopicMF) that simultaneously considers the ratings and accompanied review texts. Wu et al. [15] proposed a cyclic recommendation network to learn the implicit representation of users and items via traditional matrix factorization methods. Then these static hidden features are input into the RNN model to learn dynamic hidden features according to the order of time. Zheng et al. [19] presented a deep model to learn item properties and user behaviors from reviews. One of the networks is used to learn user behaviors, and the other one learns item properties. Lu et al. [7] proposes a multitasking learning framework that combines the probability matrix to decompose the PMF and the confrontational Seq2Seq model. Tay et al. [13] propose a review-by-review pointer-based learning scheme, named MPCN, to extract important reviews and match them in a word-by-word fashion. It enables the most informative reviews to be utilized for prediction and deeper word-level interaction. By introducing review information, these methods alleviate the data sparsity problem. However, they ignored the dynamic change of user preference and item properties over time.

2.2 Sequential Recommender

Sequential recommender considers the order of user interactions, which utilizes the previous interactions to predict the next one. Redle et al. [10] combines the Matrix Factorization and Markov Chains to learn a transition metric over time to predict the next action for a user. Hidasi et al. [3] employs the RNN-based model for sequential recommendation and proposes the session-based model. Pei et al. [8] and Wang et al. [4]. introduce attention mechanism into neural networks for the recommender. Based on [4], Li et al. [6] considers the timestamps of interactions to explore the influence of different time intervals on the next item prediction. Ying et al. [16] models the previous item and the long history item list of a user by the attention network to obtain the long and short-term preferences of the user. Usually, RNNs-based models perform well on dense datasets, but show poor performance on sparse datasets. The above sequential recommendation methods only focused on the combination of user long and short-term preferences without considering the time and semantic irregularities in user behavior.

3 Proposed Model

In this section, we first overview the architecture of our proposed model at a high-level. Then, we explain the implementation details of each layer and how the time-aware controller works in our model.

3.1 The Overview Architecture

Our proposed model consists of three important components: two parallel hybrid neural networks, and the Factorization Machine (FM)-based fusion module. One of the networks focuses on modeling user preferences, the other one captures item popularity. Specifically, the proposed neural network consists of four layers: the embedding layer, the semantic extracting layer, the dynamic feature learning layer, and the output layer. For the embedding layer, it encodes the review and time information into one-hot vectors. Then the semantic extracting layer focuses on exploiting the obtain plentiful sentiment with sequential dependencies information, by leveraging the Bi-directional Long Short-Term Memory (BiLSTM) network. After that the dynamic feature learning layer, integrating with a time-aware controller, utilizes several convolutional fitters with varying sizes, to capture the temporal dynamic of user preferences and item popularity from multiple temporal dimensions. In the output layer, the factorization machine technique is utilized to enable the user and item latent factors to interact with each other and obtain predictions. The overall architecture of the proposed model is shown in Fig. 1.

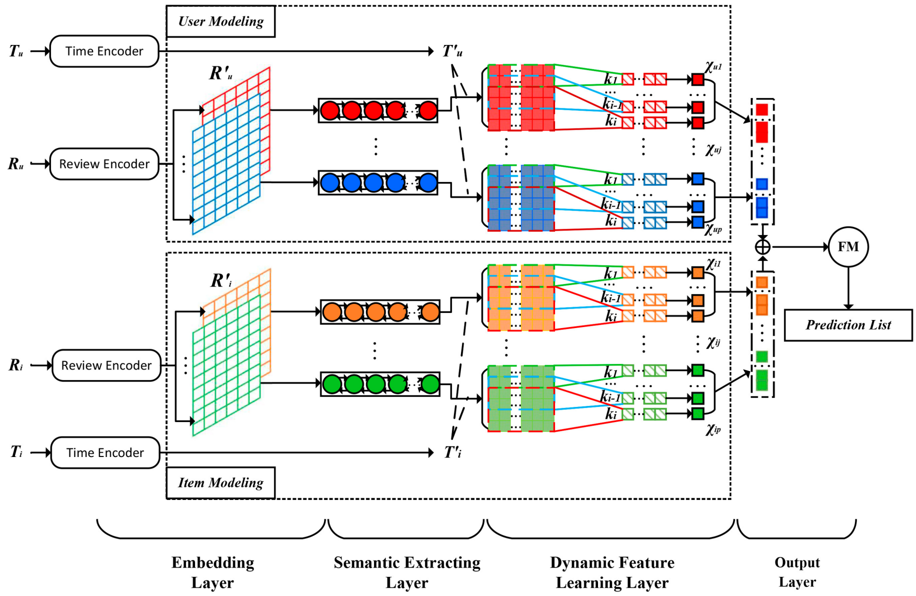

Fig. 1. Illustration of our proposed model.

3.2 The Embedding Layer

In our method, we consider reviews with time information as a strong supplementary to understand the user behavior. Therefore, review text and time from the input of the network are encoded in this layer. Let $R = \{r_1, r_2, \ldots, r_n\}$ denotes the set of n reviews of a user or an item. The i_{th} review is in the set represented as $r_i = \{w_1, w_2, \ldots, w_m\}$, where w_i represents the i_{th} word in review, and m is the number of words in this review. We type all reviews from a user or item into the embedding layer. Each review is converted into a word vector matrix through a table lookup operation from the pre-trained word matrix W_e. We use $w_i^{'} \in \mathbb{R}^d$ to represent the low-dimensional dense vectors of i_{th} word, where d is the dimension of the word vector in W_e. As a result, the $r_i^{'} \in \mathbb{R}^{m \times d}$ can be expressed as:

$$r_i^{'} = w_1^{'} \oplus w_2^{'} \oplus \cdots \oplus w_m^{'}, \tag{1}$$

where $\oplus$ denotes the concatenation operator. Therefore, the review set R is encoded as $R^{'} = [r_1^{'}, r_2^{'}, \ldots, r_n^{'}]$.

3.3 The Semantic Extracting Layer

In extraction layer, to obtain user preferences and item properties, user reviews set R_u and item reviews set R_i are input into two parallel neural networks respectively. Firstly, the words in each review are treated as sequential data. In recent, some studies have shown that Long Short-Term Memory (LSTM) performs well in processing text information. In particular, the Bi-directional LSTM (BiLSTM), an improved variant of LSTM, can capture both forward and backward contextual information. Motivated by it, we leverage the BiLSTM to obtain the sentiment information hidden in reviews.

The LSTM can change the flow of information through some gates: forgetting gate z^f , input gate z^i and output gate z^o, the equations of LSTM are as follows:

$$z^i = \sigma\left(W_i \times [h_{k-1}, x_k] + b_i\right), \tag{2}$$

$$z^f = \sigma\left(W_f \times [h_{k-1}, x_k] + b_f\right), \tag{3}$$

$$C_k = z^f \odot C_{k-1} + z^i \odot \phi\left(W_c \times [h_{k-1}, x_k] + b_c\right), \tag{4}$$

where C_k is the cell status of the k_{th} LSTM unit, h_{k-1} and x_k represent the previous hidden state and the current input respectively. Then, the output content z^o is determined based on the content saved by the cell state C_k, and the content stored in the cell state is selectively output:

$$z^o = \sigma\left(W_o \times [h_{k-1}, x_k] + b_o\right). \tag{5}$$

The hidden state of the k_{th} step is:

$$h_k = z^o \odot \phi(C_k), \tag{6}$$

where W_i,W_f,$W_o \in \mathbb{R}^{D\times D}$ are trainable parameters, and D is the dimension of input embedding and hidden layer in LSTM. In the above formulas, σ and ϕ represent the sigmoid function and tanh function, and $\odot$ denotes the elementwise product operation.

The BiLSTM contains forward and backward LSTM, which can capture the syntax and meaning of words respectively. The forward calculation of the forward layer and the inverse calculation of the backward layer are combined to obtain z_k^o:

$$\overrightarrow{h}_k = f(w_1 x_k + w_2 \overrightarrow{h}_{k-1}), \tag{7}$$

$$\overleftarrow{h}_k = f(w_3 x_k + w_4 \overleftarrow{h}_{k-1}), \tag{8}$$

$$z_k^o = g(w_5 \overrightarrow{h}_k + w_6 \overleftarrow{h}_k), \tag{9}$$

where $\overrightarrow{h}_k$ and $\overleftarrow{h}_k$ represent the hidden state of forward LSTM and backward LSTM at the k_{th} step, respectively. In the BiLSTM, the hidden state of the time step k is updated as:

$$h_k = \left[\overrightarrow{h}_k, \overleftarrow{h}_k\right]. \tag{10}$$

After the BiLSTM treatment, the massive reviews in R are turned into plentiful sentiment information set, denoted as $Y = [y_1, y_2, \ldots, y_n]$, $Y \in \mathbb{R}^{g\times p}$, where g is the length of the input review sequence from a user or item, p is the size of a review embedding. Then, we further extract the user preferences and item popularity from Y by leveraging CNN, which consists of a convolution layer and a pooling layer.

3.4 The Dynamic Feature Learning Layer

To address the time irregularity hidden in the sequence of user behaviors, we introduce the time-aware controller to make our model sensitive to time changes. We assume that the information hidden in two adjacent interactions with a short time interval is greater than two adjacent interactions with a large time interval. Hence, we employ time interval information to perceive dynamic changes in user and item characteristics. Let $T = \{t_1, t_2, \ldots, t_n\}$ denotes the set of review time in R. Furthermore, we also normalized the time interval between i_{th} review and $(i+1)_{th}$ review. Therefore, the time information of the i_{th} review is encoded as follows:

$$\tilde{t_i} = t_{i+1} - t_i, (i \in [1, n)), \tag{11}$$

$$t_i^{'} = \frac{\tilde{t_i} - min(\tilde{T})}{max(\tilde{T}) - min(\tilde{T})}, \tag{12}$$

where $\tilde{T}$ is the set of intervals between two adjacent reviews, so $min(\tilde{T})$ and $max(\tilde{T})$ denote the minimum and maximum value of interaction time interval respectively.

Meanwhile, the review with longer time intervals is given less weight in our model. Hence, the y_i with time information can be represented as:

$$y_i^{'} = y_i \times \frac{1}{t_i^{'}}. \tag{13}$$

To address the semantic irregularity issue, we introduce several convolution kernels k_* to obtain user and item features from multiple temporal dimensions. Let X_i denotes the i_{th} review, and the result of the j_{th} convolution kernel operation for X_i is as:

$$l_i = \phi\left(k_j * X_i + b_j\right), \tag{14}$$

where k_j is a convolution kernel, b_j is a bias term, the $*$ represents the convolution operation, and ϕ is an activation function. For the i_{th} review, we can get a feature set $L_j = [l_1, l_2, \ldots, l_h]$ by k_j, where h is the count of a convolution kernel operation result. In our method, we employ multiple different sized kernels for feature acquisition. After the convolution operations, the max pooling layer is introduced to get the most meaningful feature. We can formalize it as follows:

$$z_j = \max\left(L_j\right), \tag{15}$$

$$Z_i = \left[z_1, z_2, ..., z_s\right], \tag{16}$$

where z_j is the result of max pooling from L_j, s is the number of convolutional kernels, and Z_i is the set of one review features. Multiple review feature vectors are connected to model the user or item, expressed as:

$$\varUpsilon = Z_1 \oplus Z_2 \oplus ... \oplus Z_q. \tag{17}$$

Finally, we put the result of max pooling operation into a fully connected layer.

$$\chi = \delta \left(W \times \Upsilon + b \right), \tag{18}$$

where δ represents the *ReLU* function, W is a weight matrix, b is a bias term, and $\chi \in \mathbb{R}^{o \times 1}$ is a one-dimensional feature vector for a user or item. For easily distinguishing, the feature representation of a user and an item are identified as χ_u and χ_i respectively.

3.5 The Output Layer

We obtain user representation and item representation from the two above parallel neural networks. Motivated by the excellent performance of FM, we connect χ_u and χ_i into FM to obtain the corresponding prediction rating. Then, we select the top N as a list of recommendations for user. We formulate the process as follows:

$$\Psi = \chi_u \oplus \chi_i, \tag{19}$$

$$\langle v_n, v_m \rangle := \sum_{f=1}^{K} v_{n,f} \cdot v_{m,f}, \tag{20}$$

$$\hat{y} := \omega_0 + \sum_{n=1}^{N} \omega_n \Psi_n + \sum_{n=1}^{N} \sum_{m=n+1}^{N} \langle v_n, v_m \rangle \Psi_n \Psi_m, \tag{21}$$

where ω_0 is the global bias, ω_n represents the n_{th} variable strength, $\langle v_n, v_m \rangle$ represents the interaction between the n_{th} and m_{th} variables. During the training process, we utilize the loss function as follows:

$$\zeta = \gamma + \eta \parallel \Theta \parallel_2, \tag{22}$$

$$\gamma = \frac{1}{2} \sum_{i=1}^{N} \left(\hat{y_i} - y_i \right)^2 + \xi, \tag{23}$$

where ξ is the regularization term to prevent over-fitting, η and Θ is the penalty coefficient and the set of trainable parameters, respectively. In (23) , N is the number of samples, $\hat{y_i}$ is the prediction and y_i is the truth.

4 Experiments

4.1 Experimental Settings

Datasets. In our experiments, we utilize the Amazon dataset for experimental analysis. Specifically, we select three different subcategories from Amazon as subset for experiment, i.e., Musical Instruments (MIs), Automotive (Auto) and Luxury Beauty (LB). The statistics of three datasets are shown in Table 1.

Table 1. The statistics of Amazon datasets.

Datasets	#users	#items	#reviews	Density
LB	19947	1798	23799	0.07%
Auto	2928	1835	20473	0.38%
MIs	1429	900	10261	0.80%

Baseline Models. To demonstrate the superiority in our proposed model, we compare it with the following baseline models:

- **BPR** [9]: The model optimizes for a pairwise ranking objective function. Matrix factorization is introduced as a recommender.
- **GRU4Rec** [3]: This model is a session-based recommendation and RNNs are introduced to model user interaction sequences.
- **CFKG** [18]: It considers various item relations and views interaction as another relation between users and items.
- **TiSASRec** [6]: Based on the SASRec, it considers the temporal information and the relative position of the interactions.
- **SLRC** [14]: It introduces Hawkes Process into Collaborative Filtering (CF), explicitly addresses two item-specific temporal dynamics: short-term effect and life-time effect.

Evaluation Metrics. To evaluate the performance of different methods, we adopted two well-known metrics in Top-N recommendation: Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). In our experiments, we truncate the ranked list at 10 for two metrics. HR@10 counts the rates of the ground-truth items among the top 10 prediction items and NDCG@10 also considers the position. We select the latest interaction as the test set, the penultimate interaction as the validation set, and the remaining interactions are used for training. For each user, we randomly sample 100 negative items that are not interacted with by the user.

4.2 Results

Parameters Analysis. There are two deterministic parameters in our model worth exploring for optimal values: the size of latent factors ζ and the depth size d of the convolution layer. The performance of different values for two deterministic parameters is shown in Fig. 2. As we can see, the optimal values of ζ and d should be assigned to 50 and 3 respectively. We also test the dimension of word embedding κ with the range of [100, 200, 300, 400, 500], the batch size b in [32, 64, 128, 256], the learning rate λ in [0.0001, 0.0005, 0.001, 0.005]. According to the performance and efficiency of our proposed model, the κ, b and λ are set as 300, 128 and 0.001. The performance decreases when the batch size is below this optimal value because too little data does not have the characteristics of the overall sample. However, the batch size value is too large to increase memory usage and reduce efficiency. The value of learning rate mainly depends on the stability and time consuming of the model. We implement our proposed model with *PyTorch*.

Effectiveness of Time-Aware Controller. To verify the effect of time-aware controller on modeling user and item, we then conduct several experiments with three variants of our model:

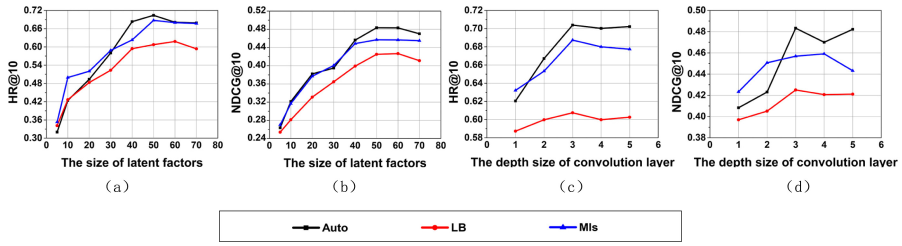

Fig. 2. Performance as a result of varying ζ shown in (a) and (b), and performance as a result of varying d shown in (c) and (d).

- **Ours-NoT:** Both timestamp information and time-aware controller are not used in this variant.
- **Ours-UT:** We introduce the time-aware controller in the users preferences net, which is equivalent to only considering the dynamic preferences of the user.
- **Ours-IT:** Similar to Ours-UT, instead of introducing a time-aware controller for item p, the variant utilizes it for modeling items to explore the dynamic properties over time.

The performance of our model and its three variants are shown in Table 2. As shown, the proposed model delivers the best results, and the Ours-UT and Ours-IT perform better than the Ours-NoT, which demonstrates the efficiency of time-aware controller for capturing dynamic preferences of users and items.

Comparison to Baselines. The performance of our model and all baselines are shown in Table 3. As shown, SLRC performs better compared with other baselines. Our model performs best on three datasets among all the recommendation models in our experiments, which indicates the proposed model can better capture dynamic characteristics of users and items. We can summarize the reasons as follows: (1) user reviews, as the complementary data source, contain more information than ratings. From all the historic reviews, we can extract user long-term preferences and static inherent properties of items. (2) we combine

Table 2. The performance of our proposed model and its variants.

Models	Ours-NoT	Ours-UT	Ours-IT	Ours
Luxury beauty				
HR@10	0.5331	0.5601	0.5877	**0.6075**
NDCG@10	0.3699	0.3720	**0.4261**	0.4251
Automotive				
HR@10	0.5964	0.6981	0.6653	**0.7039**
NDCG@10	0.4021	0.4762	0.4770	**0.4833**
Musical instruments				
HR@10	0.6003	0.6713	0.6621	**0.6874**
NDCG@10	0.3913	0.4231	0.4356	**0.4569**

BiLSTM and CNN in the two parallel networks. BiLSTM is capable of obtaining the sentiment information existing in review texts, and CNN can further extract the features of user and item. (3), we employ a time-aware controller to capture the short-term preferences of the user and item features.

Table 3. Models performance. The best result in each row is boldfaced, and the second best one in each row is underlined.

Models	Luxury beauty		Automotive		Musical instruments	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
BPR	0.4620	0.2917	0.4668	0.3102	0.4551	0.3007
GRU4Rec	0.4857	0.2811	0.4630	0.3095	0.4989	0.3336
CFKG	0.5021	0.3383	<u>0.5945</u>	0.3915	0.5548	0.3890
TiSASRec	0.5185	0.3318	0.5023	<u>0.4137</u>	0.5240	0.3906
SLRC	<u>0.5243</u>	<u>0.3907</u>	0.5662	0.3871	<u>0.5891</u>	<u>0.4038</u>
Ours	**0.6075**	**0.4251**	**0.7039**	**0.4833**	**0.6874**	**0.4569**

5 Conclusion

In this paper, we propose a novel neural network-based model for sequential recommendation, which can capture the dynamic preferences of users and items in the sequence of user-item interactions. We notice that the hardness of modeling user interests from the spare implicit feedback, thus we focus on exploiting plentiful semantic information from reviews via leveraging BiLSTM. Furthermore, we observe that user's behaviors are more complex in the field of recommender system than the sequences in NLP domain, thus we further propose the time-aware controller and integrate into the CNN-based dynamic feature learning layer, which learns the user and item feature from multiple temporal dimensions. Based on these methods, it makes the proposed model more suitable for modeling user behavior. Finally, we conduct a large number of experiments on the industrial dataset to explore the performance of our proposed model, the results demonstrate that our proposed model outperforms state-of-the-art models consistently.

Acknowledgments. This research is supported in part by the Chongqing Science and Technology Bureau under grant cstc2019jscx-zdztzxX0019 and cstc2018jszx-cyzdX0041, in supported by the National Natural Science Foundation of China (No. 61902370).

References

1. Bao, Y., Fang, H., Zhang, J.: TopicMF: simultaneously exploiting ratings and reviews for recommendation. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, p. 2C8. AAAI 2014, AAAI Press (2014)
2. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. Comput. Sci. (2014)
3. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks (2015)
4. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation, pp. 197–206 (2018)
5. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
6. Li, J., Wang, Y., Mcauley, J.: Time interval aware self-attention for sequential recommendation. In: WSDM 2020: The Thirteenth ACM International Conference on Web Search and Data Mining (2020)
7. Lu, Y., Dong, R., Smyth, B.: Why I like it: multi-task learning for recommendation and explanation. In: the 12th ACM Conference (2018)
8. Pei, W., Yang, J., Sun, Z., Zhang, J., Bozzon, A., Tax, D.: Interacting attention-gated recurrent networks for recommendation, pp. 1459–1468 (2017)
9. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009 (2012)
10. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation, pp. 811–820 (2010)
11. Shi, X., Luo, X., Shang, M., Gu, L.: Long-term performance of collaborative filtering based recommenders in temporally evolving systems. Neurocomputing **267**, 635–643 (2017)
12. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding (2018)
13. Tay, Y., Tuan, L., Hui, S.: Multi-pointer co-attention networks for recommendation (2018)
14. Wang, C., Zhang, M., Ma, W., Liu, Y., Ma, S.: Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In: The World Wide Web Conference (2019)
15. Wu, C.Y., Ahmed, A., Beutel, A., Smola, A., Jing, H.: Recurrent recommender networks, pp. 495–503 (2017)

16. Ying, H., Zhuang, F., Zhang, F., Liu, Y., Wu, J.: Sequential recommender system based on hierarchical attention networks. In: Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18 (2018)
17. Yu, Z., Lian, J., Mahmoody, A., Liu, G., Xie, X.: Adaptive user modeling with long and short-term preferences for personalized recommendation. In: Twenty-Eighth International Joint Conference on Artificial Intelligence IJCAI-19 (2019)
18. Zhang, Y., Ai, Q., Chen, X., Wang, P.: Learning over knowledge-base embeddings for recommendation (2018)
19. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: The Tenth ACM International Conference (2017)