CSAN: Contextual Self-Attention Network for User Sequential Recommendation

Xiaowen Huang[1,2], Shengsheng Qian[1], Quan Fang[1], Jitao Sang[3,4], Changsheng Xu[1,2]
[1]National Lab of Pattern Recognition, Institute of Automation, CAS, Beijing 100190, China
[2]University of Chinese Academy of Sciences
[3]School of Computer and Information Technology & Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University
[4]State Key Laboratory for Novel Software Technology, Nanjing University
xiaowen.huang77@gmail.com,{shengsheng.qian,qfang}@nlpr.ia.ac.cn,jtsang@bjtu.edu.cn,csxu@nlpr.ia.ac.cn

ABSTRACT

The sequential recommendation is an important task for online user-oriented services, such as purchasing products, watching videos, and social media consumption. Recent work usually used RNN-based methods to derive an overall embedding of the whole behavior sequence, which fails to discriminate the significance of individual user behaviors and thus decreases the recommendation performance. Besides, RNN-based encoding has fixed size and makes further recommendation application inefficient and inflexible. The online sequential behaviors of a user are generally heterogeneous, polysemous, and dynamically context-dependent. In this paper, we propose a unified **C**ontextual **S**elf-**A**ttention **N**etwork (**CSAN**) to address the three properties. Heterogeneous user behaviors are considered in our model that are projected into a common latent semantic space. Then the output is fed into the feature-wise self-attention network to capture the polysemy of user behaviors. In addition, the forward and backward position encoding matrices are proposed to model dynamic contextual dependency. Through extensive experiments on two real-world datasets, we demonstrate the superior performance of the proposed model compared with other state-of-the-art algorithms.

KEYWORDS

self-attention; contextual; sequential recommendation; multi-modal

ACM Reference Format:
Xiaowen Huang, Shengsheng Qian, Quan Fang, Jitao Sang, Changsheng Xu. 2018. CSAN: Contextual Self-Attention Network for User Sequential Recommendation. In 2018 ACM Multimedia Conference (MM'18), October 22-26, 2018, Seoul, Republic of Korea. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240609

1 INTRODUCTION

With the rapid development of the Internet, some applications of sequential scenario have become pervasive and multilateral, such as news feed consumption, product purchase recommendation and ad click prediction. A user's behaviors in such applications are a sequence in chronological orders, and his subsequent behaviors can be predicted by sequential recommendation methods. Modeling and predicting the complex sequential interactions between users and items is very important for providing personalized services in recommendation systems [32].

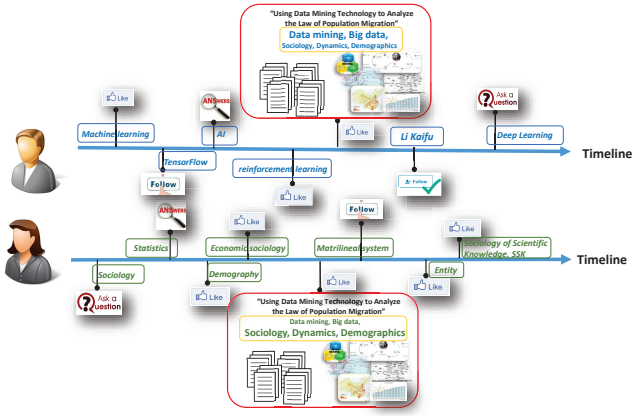

Figure 1: A schematic diagram of the behavior sequence of two users. Text describes the content topics of user actions at different timestamps. Red rectangles show the two users' different attentions on the same article due to their different contextual behaviors.

Regarding the importance of modeling user sequential behaviors, considerable work has been conducted to make sequential recommendations with user historical records in recent years [18, 26]. The Markov Chain (MC) based model and Matrix Factorization (MF) based approach [3] are the early approachs to sequential recommendation [10, 26]. However, both of these two methods are inadequate that MC combines all the components independently [31] and MF suffers from sparsity issues [11]. Inspired by the great success of the neural network, RNN with sequential architecture capturing long-range dependencies, and CNN whose hierarchical structrue is good at extracting local or position-invariant features, have been successfully employed to encode the sequence for different applications, such as sequential click prediction [36] and top-N sequential recommendation [29]. However, RNN or CNN based methods tend to compress all of a user's previous records into a fixed hidden representation, which ignores the individual importance of user

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240609

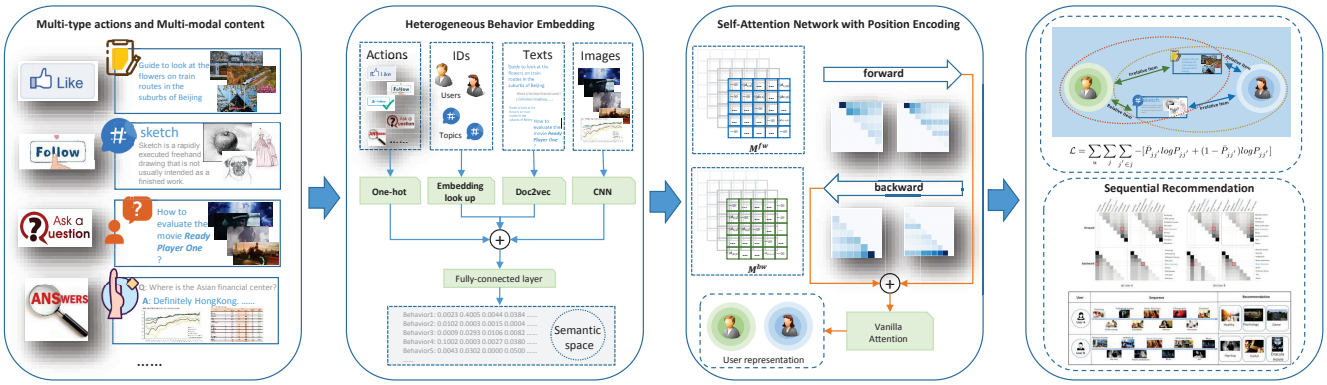

Figure 2: The schematic illustration of the sequence modeling architecture.

behaviors and might impair user interest modeling [29]. Besides, the sequential nature of RNN makes parallelization challenging [4]. Recently, Vaswani et al. argue that self-attention mechanisms [30], without using either recurrence or convolution, called *Transformer*, can be effective in sequence-to-sequence modeling tasks, which is more flexible in sequence length than RNN/CNN, and is more task/data-driven when modeling dependencies. ATRank [38] takes the lead in using self-attention structure to model user behavior for the sequential recommendation and achieves encouraging results.

However, modeling user sequential behavior in real-world toward recommendation is still a challenging problem by exploring users' history records. The user sequential behaviors exhibit certain distinctive characteristics: (1) User behaviors are inherently **heterogeneous**, including diversity of actions and multi-modal property of content. Rather than with merely one type of action, such as clicking on websites, there are many sequential scenarios with multiple types of actions towards items. For example, as Fig.1 shows, users act on items in a wide variety of ways, e.g., following, voting, asking/answering the questions on Q&A communities. Different actions reflect a different degree of importance to the item. For instance, comparing with voting an answer to a specific question, adding a user's own answer directly to this question can better represent user's attention to this question. Most of the research work focuses on modeling single-type action [9, 29], which cannot be handled well for the different degrees of user interest. In addition, the items consist of multi-modal data, such as textual content (e.g., tweets, comments) and visual content (e.g., photos and videos) which may provide valuable information for item representation. Leveraging all available information about users and items can learn more accurate user preferences. (2) User behaviors have the **polysemy** property. Since the word embedding usually suffers from polysemy in natural language, an action may have a variety of meanings in different contextual scenarios. For example, as the two users shown in Fig. 1, they all like the same article named "Using Data Mining Technology to Analyze the Law of Population Migration", which is on the topics "Data mining", "Big data", "Sociology", "Dynamics", "Demographics". But the focus of male user could be "Data mining", "Big data" because his context behavior is artificial intelligence technology, but the female user may be interest in "Sociology", "Dynamics", "Demographics" because her context is on sociology. Thus, it is important to consider the meaning of the same behavior in different contexts for building a personalized sequence recommendation system. However, the existing methods rarely consider this problem. (3) User behaviors are **dynamically context-dependent**. A better recommender should capture both the long-term user taste and short-term sequential effect [26]. On one hand, how to model the temporal relationship of user behaviors remains an open problem. On the other hand, how to improve model efficiency and speed up personalized recommendation is also questionable. To the best of our knowledge, the idea to consider heterogeneity, polysemy and the dynamic contextual dependency of user behaviors in a principled way for personalized sequential recommendation is unexplored and challenging.

In this paper, we aim to tackle above issues by introducing a new user sequence modeling framework, Contextual Self-Attention Network (CSAN), based solely on self-attention for sequential recommendation by exploring user heterogeneous sequential behaviors. The framework is shown in Fig. 2. The input consists of multi-types actions. Each item that user interacts with consists of multi-modal content. The main advantages of CSAN are: 1) To deal with the heterogeneity of user behaviors, we develop a heterogeneous behavior embedding network to model multiple types of actions and multiple modalities of items by projecting the user's original behaviors into a common semantic space. 2) To capture the polysemy of user behaviors, a feature-wise self-attention is processed to extract different aspects of the sequence to model the complicated correlations. Since attention on different features contains different information about dependency in flexible contexts, it is able to handle the variation of context around the same behavior. Benefiting from discriminating the significance of different user behaviors, the feature-wise self-attention network can strengthen the recommendation performance. 3) The modeling of the dynamical contextual dependency is accomplished by our position encoding matrices instead of complex RNN/CNN structure, or timestamp encoding in ATRank. This method can easily encode prior structure knowledge such as temporal order and dependency parsing to model the asymmetric attention between two elements [27]. Through the position encoding strategy, long-term dependency and short-term interest of the user can be captured. Moreover, the proposed system can reduce the training time and improve the recommended efficiency. As a result, each user is modeled as a compact representation considering

the heterogeneous, contextual, and dynamical behaviors. The sequential recommendation is conducted by downstream application networks.

The main contributions can be summarized as follows:

- We propose a novel contextual self-attention network for the sequential recommendation, which can leverage user historical behaviors in a more effective manner and have high computational efficiency.
- We propose to employ embedding network, self-attention mechanism and position encoding to deal with the heterogeneity, polysemy, and dynamic contextual dependency of user sequential behaviors. This can accurately capture the user's interests and critical information for the sequential recommendation.
- Extensive experimental results on both singe-type behavior dataset and multi-type multi-modal behavior dataset demonstrate the superior performance of the proposed model compared with other state-of-the-art algorithms. In addition, we introduce a multi-type and multi-modal behaviors dataset, and we will release the dataset for research purpose[1].

2 RELATED WORK

Sequential Recommendation Sequential recommendation problem is usually cast as sequence prediction problem. Most existing approaches focus on Markov Chain (MC) based methods and Neural network-based methods. Scalable sequential models usually rely on MC to capture sequential patterns [7, 26], where an L-order Markov chain makes recommendations based on L previous actions. However, a major problem of MC based models is that all the components are independently combined, indicating that it makes strong independence assumptions among multiple factors [31]. Recently, a Matrix Factorization (MF) based approach factorizes the matrix of transition probability from the current item to the next one into the latent factors [3]. However, MF easily suffers from sparsity issues due to the power-law distributed data in the real word [11]. Inspired by the great power of matrix factorization, Factorized Personalized Markov Chain (FPMC) [26] combines the power of MF and MC to factorize the transition matrix over underlying MC to model personalized sequential behaviors for the next-basket recommendation. FPMC and its variant [2] improve this method by factorizing this transition matrix into two latent and low-rank sub-matrices. All the MC-based methods have the same deficiency that these recommenders only obtain the local sequential behaviors between every two adjacent items.

Recently, Recurrent Neural Network (RNN) approaches have achieved much success in sequence modeling [22]. It has been successfully employed to model temporal dependency for different applications, such as sentence modeling tasks [22–24], video modeling [6], sequential click prediction [36], multi-behavioral sequential prediction [19] and location prediction [20]. Though it is an efficient way to encode user context, it still suffers from several difficulties, such as hard to parallelize, time-consuming, hard to preserve long-term dependencies. Recently, Convolutional Neural Network (CNN) based encoding methods have also achieved comparable performance with RNN in many sequence processing tasks [14]. CNN structures are exploited to encode the behavior sequence for downstream applications [29]. Both the basic RNN and CNN encoders suffer from the problem that the fixed-size encoding vector may not support both short and long sequences well.

Attention Mechanism It is quite challenging to learn the relevance of items and context. The attention mechanism [1] is introduced to provide the ability to refer specific records dynamically in the decoder, which has already achieved great successes in fields like reading comprehension [5, 17], ads recommendation [35, 39] and computer vision [34] in recent years. Self-attention are also studied in different mechanisms [5, 17, 30], in which inner-relations of the data at the encoder side are considered. The *Transformer* [30] solely uses attention mechanism to construct a sequence to sequence model that achieves a state-of-the-art quality score on the neural machine translation task. ATRank [38], which based only on self-attention, is proposed for recommendation tasks through projecting all types of behaviors into multiple latent semantic spaces where influence can be made among the behaviors via self-attention.

Our work differs from the above approaches in that we introduce a unified RNN/CNN-free user behavior modeling framework based solely on self-attention for sequential recommendation whose attention mechanism works on the feature level instead of element level, and use position encoding matrices to model dynamic contextual dependency instead of time encoding. This is effective and can improve the performance in our task.

3 OUR PROPOSED METHOD: CSAN

As shown in Figure 3[2], for addressing the sequential recommendation problem, our network consists of several major blocks.

3.1 Problem Statement

Before going into the details of our proposed model, we first define the problem and basic concepts. General user behaviors can be interpreted using the binary relationship between a user and an item. We denote $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$ as the set of users, $\mathcal{A} = \{a_1, a_2, ..., a_{|\mathcal{A}|}\}$ and $\mathcal{I} = \{i_1, i_2, ..., i_{|\mathcal{I}|}\}$ are the sets of user actions and items, where $|\mathcal{U}|$, $|\mathcal{A}|$, $|\mathcal{I}|$ denotes the number of elements in the set of User, Action and Item, respectively. Each behavior of user u is associated with a timestamp, hence each user behavior can be formulated as a tri-tuple $\{a, i, t\}$, where a stands for an action the user takes, i is a item that the action works on, and t is the timestamp when the behavior happens. Finally a user can be represented as his/her sequential behaviors $u = \{a_{t_j}, i_{t_j}, t_j | j = 1, 2, ..., |J|\}$. Given a behavioral history of a user towards items, the task is to predict the next item that the user may act on.

3.2 Heterogeneous Behavior Embedding

A heterogeneous behavior network contains multiple types of actions and multiple modalities of items. Due to the heterogeneity of behaviors, it is necessary to map the original features to a common semantic space.

3.2.1 Multi-type actions representation. For each user u, given a specific item i, there is an interaction between the user and the item at time t, called an action $a^t_{u,i}$. Different actions on the same item indicate the difference degree of attention of the user attaches to the item. Therefore, it is necessary to give a unique representation

[1] https://acmmm2018csan.github.io/

[2] '+','R','·' in the model means 'concat','or', and 'dot-product', respectively.

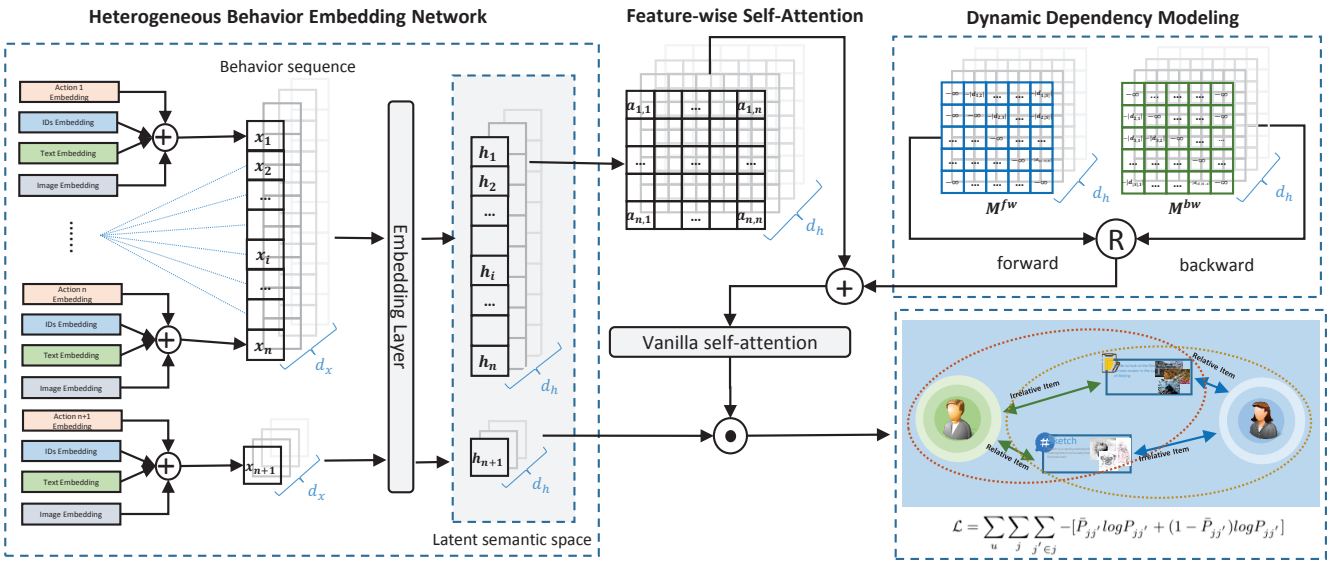

Figure 3: Illustration of the Contextual Self-Attention Network (CSAN) model.

for each kind of action. We perform a direct one-hot representation $\{vec(a_m) \in \mathbb{R}^{|A|} | m = 1, 2, ..., |A|\}$ on the actions, trying to capture the impacts of different behavioral types.

3.2.2 Multi-modal content representation. We learn both the textual and visual features to enhance and complement each other for better item representation. For the textual features, we utilize a widely-used Doc2vec [16] model to learn textual semantic representation, which is represented as: $vec(item_i)^{text} \in \mathbb{R}^{64} | i \in |I|)$. Inspired by the visual feature representation ability of deep architecture, we employ AlexNet pre-trained on ImageNet to extract visual semantic features. Due to the possible redundancy and noise of visual information, we reduce the high dimensional deep feature to 64 ones with PCA [33] to reduce noise interference and improve efficiency. Finally, the image features will be represented as: $vec(item_i^{img} \in \mathbb{R}^{64} | i \in |I|) = \frac{1}{|K|} \sum_{k \in K} vec(image_k)$, where $|K|$ denotes the total number of images included in the item.

3.2.3 Semantic space embedding. By integrating multiple features, user behavior u_i for user u acts on item i is defined as:

$$u_i = vec(a_{u,i}) + lookup_{u,i}(IDs) + vec(item_{u,i}^{text}) + vec(item_{u,i}^{image})$$

where $lookup_{u,i}(IDs)$ represents that we perform a direct lookup on the user and item IDs. Here " + " denotes concatenation.

Arranging items in chronological order, we can get the user's behavior sequence $S^u = \{u_1, u_2, ..., u_{t_{|S|}}\} \in \mathbb{R}^{d_x \times |S|}$. d_x is the number of dimensions, $|S|$ is the length of the behavior sequence. After integration, an embedding layer is used to project the raw features into the latent semantic space. In our model, we transform the input sequence S^u to a sequence of hidden state $H^u = \{h_1^u, h_2^u, ..., h_{|S|}^u\} \in \mathbb{R}^{d_h \times |S|}$ by a fully-connected layer,

$$H = \sigma_h(W_h S + b_h) \tag{1}$$

where W_h and b_h are the learnable parameters, and $\sigma_h(\cdot)$ is an activation function.

Heterogeneous behavior information can be effectively integrated by the embedding layer. It is helpful to construct a more comprehensive user portrait by considering the diversity of actions and the multimodal property of the content.

3.3 Feature-wise Self-attention

To model the 'polysemous behaviors', we employ a feature-wise self-attention mechanism, which is a natural extension of additive attention [1] at feature level [27]. Through the mapping of the semantic space network described above, we get a semantic representation of each behavior with the same dimension. Instead of computing a single scalar score for each element h_j^u, the feature-wise self-attention computes a feature-wise score vector for h_j^u by replacing the weight value with a self-attention matrix. The correlations between h_i and h_j which are from the same sentence is:

$$f(h_i, h_j) = W^T \sigma(W_1 h_i + W_2 h_j) \tag{2}$$

where the weight score $f(h_i, h_j) \in \mathbb{R}^{d_h}$ is a vector with the same length as h_i, and all the weight matrices $W, W_1, W_2 \in \mathbb{R}^{d_h \times d_h}$. Then the probability value between h_i and h_j is computed at feature level:

$$a_{ij}^k = \frac{e^{[f(h_i, h_j)]_k}}{\sum_{j=1}^{|S|} e^{[f(h_i, h_j)]_k}} \tag{3}$$

where $k \in \{1, 2, ..., d_h\}$. The weight value on each feature is normalized along corresponding feature dimension, which ensures that the correlations for elements are modeled at feature level.

After attention scores are computed over all elements, the output for h_j can be defined as:

$$e_j^k = \sum_{i=1}^{|S|} a_{ij}^k \odot h_i^k \tag{4}$$

The attention mechanism takes the whole hidden states $H^u = \{h_1^u, h_2^u, h_{|S|}^u\}$ as input, and outputs the context-aware representation $E^u = \{e_1^u, e_2^u, e_{|S|}^u\}$. By computing a score for each feature of

each behavior through feature-wise self-attention, we can better capture the different contribution scale of various contextual items and select the features that can best describe the behavior's specific meaning in any given context, to deal with the problem of behavioral polysemy.

3.4 Dynamic Dependency Modeling

In order to incorporate information about the temporal order of the sequence and model dynamic contextual dependencies, the position of the elements should be taken into consideration. In our work, we propose two *Position Encoding Matrices* in forward and backward to encode the position information for constructing the dynamic dependency. The forward and backward matrices are defined as:

$$M^{fw} = \left\{ \begin{bmatrix} -\infty & -|d_{1,2}| & \cdots & \cdots & -|d_{1,|S|}| \\ -\infty & -\infty & -|d_{2,3}| & \cdots & -|d_{2,|S|}| \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & -|d_{|S|-1,|S|}| \\ -\infty & \cdots & \cdots & \cdots & -\infty \end{bmatrix} \right\}_{k=1}^{d_h} \tag{5}$$

$$M^{bw} = \left\{ \begin{bmatrix} -\infty & \cdots & \cdots & \cdots & -\infty \\ -|d_{2,1}| & -\infty & \cdots & \cdots & -\infty \\ -|d_{3,1}| & -|d_{3,2}| & -\infty & \cdots & -\infty \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -|d_{|S|,1}| & \cdots & \cdots & -|d_{|S|,|S|-1}| & -\infty \end{bmatrix} \right\}_{k=1}^{d_h} \tag{6}$$

where the matrix $M^{fw/bw} \in \mathbb{R}^{|S|\times|S|\times d_h}$, $k = \{1, 2, ..., d_h\}$ denotes feature index. Position encoding also works on feature-level. Taking into account the relative position of the two elements, the positional deviation between i and j, $|d_{i,j}|$, are exploited.

In the forward encoding process, M_{ij}^{fw} equals to $-|d_{i,j}|$ when the element i is earlier than element j, $-\infty$ otherwise. And vice versa in the backward encoding process. Note that, there are several approaches for positional encoding, including the sinusoidal functions in [30], the diagonal-disabled mask in [12], the bi-directional masks for obtaining asymmetric attention matrix in [27], and other forms [8, 28]. However, the proposed position encoding strategy is highly effective in our tasks.

Unlike *Transformer*, neither stacking of attention blocks nor an encoder-decoder structure is required. To this end, we add position encoding matrices to the feature-wise attention architecture. The attention weight with position encoding can be modified as:

$$f^*(h_i, h_j) = W^T \sigma(W_1 h_i + W_2 h_j) + repmat(M_{ij}, d_h) \tag{7}$$

where $repmat(m, n)$ is a function for replicating the value m to n-dimensional vector. The updated probability value is:

$$a_{ij}^{*k} = \frac{e^{[f(h_i,h_j)]_k + (-|d_{ij}|)}}{\sum_{j=1}^{|S|} e^{[f(h_i,h_j)]_k + (-|d_{ij}|)}} = e^{-|d_{ij}|} \cdot \frac{e^{[f(h_i,h_j)]_k}}{\sum_{j=1}^{|S|} e^{[f(h_i,h_j)]_k + (-|d_{ij}|)}} \tag{8}$$

where $M_{ij} = -|d_{i,j}|$ and $M_{ji} = -\infty$, a_{ij}^{*k} decreases with the increase of $|d_{ij}|$ which models the impact of context at different positions. $a_{ji}^{k} = 0$ means no attention of element i to element j.

After forward and backward procedures, the output $E_u^{fw} = \{\mathbf{e}_1^{fw}, \mathbf{e}_2^{fw}, ..., \mathbf{e}_{|S|}^{fw}\}$ and $E_u^{bw} = \{\mathbf{e}_1^{bw}, \mathbf{e}_2^{bw}, ..., \mathbf{e}_{|S|}^{bw}\}$ are concatenated vertically to a $2d_h$ by $|S|$ matrix. Then a vanilla self-attention layer is used to map the combination matrix to a vector $Seq^u \in \mathbb{R}^{d_h}$. The final representation combines a temporal order encoded and context-aware vector representation for each independent behavior, which considers behavioral polysemy and dynamic dependency including long-term user taste and short-term interest. The simple architecture leads to fewer parameters, less computation and easier parallelization.

3.5 A unified sequential recommendation Framework

In several recent results from the deep learning community, it has been observed that joint inference with multiple related tasks can lead to superior performance in each of the individual task, while drastically improves the training behavior. Hence, we use a loss function that jointly evaluates the performance of all tasks, which can be expressed as follows:

Given a user and the associated historical sequence, the ultimate goal of the task is to rank the ground-truth item j higher than all other items ($j^{'} \in \mathcal{I} \backslash j$). Therefore it is a natural choice to optimize the pairwise ranking between j and $j^{'}$. For an item j to be predicted, the prediction probability $P_{jj'} = p(j >_{u_h} j^{'})$, where $>_{u_h}$ means given the user's historical behaviors u_h, item j is ranking ahead of $j^{'}$. $\bar{P}_{jj'}$ represents the ground-truth probability. The final loss function is the cross entropy loss:

$$\mathcal{L} = \sum_u \sum_j \sum_{j' \in j} -[\bar{P}_{jj'} log P_{jj'} + (1 - \bar{P}_{jj'}) log P_{jj'}] \tag{9}$$

where $P_{jj'} = \sigma_{jj'} = \sigma(f(j) - f(j^{'}))$, $f(\cdot)$ denotes the ranking function who can be a dot-product function or a more complex deep neural network. $\sigma(\cdot)$ is the sigmoid function.

4 EXPERIMENTS

4.1 Evaluation Datasets

We evaluate our proposed method on two real-world datasets: Amazon product dataset and Zhihu activity dataset. Amazon[3] is an *e-commerce website* where users interacts with the commodity. Zhihu[4] is a Chinese *question-and-answer website* where questions are created, answered, edited and organized by the community of its users.

Amazon product dataset[5] The Amazon product dataset, comprising large corpora of reviews and timestamps on various products, spanning May 1996 to July 2014, was recently introduced by [21]. In this paper, we take a series of large categories including 'Automotive', 'Office Products', 'Toys and Games', 'Clothing, Shoes and Jewelry', and 'Video Games' for experiment. This set of data is notable for its high sparsity and variability. We retain some of the characteristics used to construct the user sequential behaviors.

Zhihu activity dataset To collect multi-type behaviors dataset for the sequential recommendation, we crawl Zhihu users' dynamic activities data. Starting with a specific user, we finally collect 10458 users through their following and follower lists layer by layer. For each user, we crawl his/her dynamic activities for one year, including multi-type actions and multi-modal content. We choose 6

[3] https://www.amazon.com/
[4] https://www.zhihu.com/
[5] http://jmcauley.ucsd.edu/data/amazon/

Table 1: Statistics of Amazon product dataset

Sub-dataset	#Users	#Items	#Categories	#Samples
Automotive	851418	320112	1960	20473
Office	909314	130006	662	53258
Toys	1342911	327698	614	167517
Clothing	3117268	1136004	1051	278677
Games	826767	50210	83	231780

Table 2: Statistics of Zhihu activity dataset

#Users	#Items	#Topics	#Samples	#Images
10458	426821	5000	644767	100779

most frequent actions including following question/topic, creating question/answer, and voting answer/article.

The statistics of the Amazon and Zhihu dataset are shown in Table. 1 and Table. 2, respectively. On both websites, each behavior of user has a timestamp when that behavior happens. Given the user's behavior sequence $u = (u_{t_1}, u_{t_2}, ..., u_{t_n})$ according to the chronology of the user's behavior, we make the first k behaviors of user u to predict the $(k+1)-th$ behavior in the training set, where $k = 1, 2, ..., n-2$, and we use the first $n-1$ behaviors to predict the last one in the test set.

4.2 Evaluation Metrics

As practical recommender systems usually generate a ranked list of items for a given user, we evaluate the ranking performance. AUC [25], the area under the ROC curve, is a commonly used metric for evaluating the quality of a ranking list. We report the performance of each method on the test set on both Amazon datasets and Zhizhu dataset in terms of the following ranking metrics:

$$\textbf{AUC} = \frac{1}{\mathcal{U}} \sum_{u \in \mathcal{U}} \frac{1}{|J||J'|} \sum_{j \in |J|} \sum_{j \in |J'|} \delta(p_{u,j} > p_{u,j'}) \tag{10}$$

where J denotes the positive samples set, and J' means negative. $\delta(p_{u,j} > p_{u,j'})$ is an indicator function which returns 1 if $(p_{u,j} > p_{u,j'})$ is true, and 0 otherwise. $p_{u,j}$ is the predicted probability that a user $u \in \mathcal{U}$ may act on i in the test set. A higher value of AUC indicates better performance for ranking performance. The floor of AUC from random guess is 0.5 and the best result is 1.

In order to compare the performance of all algorithms in different ways, we add another evaluation metric Precision@K on our multi-type and multi-modal Zhihu dataset.

$$\textbf{Precision@K} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \left(\frac{\sum_{j=1}^{K} pc(j, g(u))}{K}\right) \tag{11}$$

where j denotes the predict item, $g(u)$ represents the ground-truth items associated with user u we prepare to predict. $pc(j, g(u))$ is an indicator function that returns 1 if j is in $g(u)$, 0 otherwise. K is the truncation level.

4.3 Comparative Approaches

- **BPR** Bayesian personalized ranking [25] is a state-of-the-art item recommendation model which takes Matrix Factorization as the underlying predictor. It is a pairwise ranking framework.
- **CNN** Convolutional Neural Network is widely used in image recognition [13] [15] and natural language processing [14]. In recent years, CNN-based methods are used for encoding the user history as in [37]. We use a CNN structure with max pooling to model the user sequence behaviors.
- **Bi-LSTM** Long short-term memory (LSTM) units are building unit for the layer of a recurrent neural network (RNN), which are used to capture sequential dependencies and make predictions [36]. We use Bi-directional LSTM network for better capturing the forward and backward context information.
- **Bi-LSTM + Attention** Vanilla attention is added on top of the Bi-LSTM method mentioned above.
- **ATRank** ATRank [38] is an attention-based user behavior modeling framework. The method projects user behavior representations into multiple latent spaces and then self-attention mechanism is used to model the influences brought by other behaviors. The output of attention layer with "next-item" vector is fed into the downstream application network.

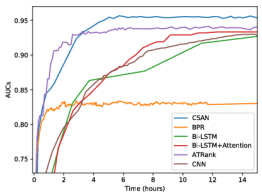

Figure 4: Convergence: Test AUC progress on Zhihu dataset.

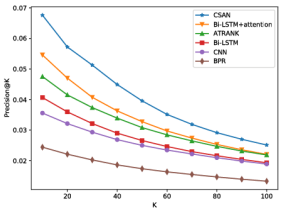

Figure 5: Precision@K on Zhihu dataset.

4.4 Implementation Details and Convergence

We implement CSAN with open-source deep learning framework Tensorflow. The whole model is trained with stochastic gradient descent (SGD) in an end-to-end way. We set the learning rate 1, batch size 32 and weight decay 0.00005. The units of latent semantic space layer are determined with cross-validation, and we find 128 is good enough.

We demonstrate the average user AUC for the test set evolves on Zhihu dataset. As shown in Fig. 4, we can see that our proposed CSAN converges faster than CNN and RNN-based methods due to the non-use of recurrent structure. It brings the possibility of parallel computing, which can greatly improve the operation efficiency. Although ATRank converges slightly faster than CSAN, it is less accurate compared with CSAN method. As for BPR, though it is the fast method of training, it has a poor performance. **Combined with accuracy and training speed, CSAN performs better than other methods.**

4.5 Performance Analysis

In this subsection, we conduct experiments on two real-world datasets. Note that except BPR and ATRank, other baselines are all based on deep neural networks since we aim to compare our approach with state-of-the-art deep learning models. Additionally, ATRank method is also based only on attention model with no RNN/CNN structure.

4.5.1 Amazon. The experimental results of all the methods for Amazon datasets are illustrated in Table 3. The main findings are

Table 3: Ranking results on Amazon datasets and Zhihu dataset (higher is better). The best performance in each case is highlighted.

Dataset	Sub-datasets	BPR	CNN	Bi-LSTM	Bi-LSTM+Att.	ATRank	CSAN-(Att.+Pos.)	CSAN
Amazon	Automotive	0.76	0.7952	0.8024	0.8054	0.7982	0,707	**0.8096**
	Office	0.8397	0.8595	0.8653	0.8646	0.858	0.8332	**0.8676**
	Toys	0.8081	0.8502	0.8616	0.8638	0.859	0.8231	**0.8692**
	Clothing	0.8127	0.8215	0.8354	0.8365	0.8317	0.787	**0.844**
	Games	0.8739	0.902	0.9062	0.9048	0.8889	0.8727	**0.9065**
Zhihu	-	0.834	0.93	0.9303	0.942	0.936	0.9261	**0.956**

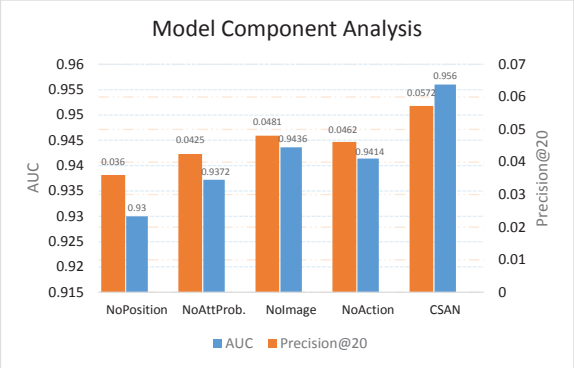

Figure 6: Experimental results on Zhihu dataset after removing the corresponding components.

summarized as follows: (1) BPR performs the worst among all methods. This is because BPR is a Bayesian-based pairwise ranking method, but user's dynamic behaviors are generally highly time-sensitive. (2) Deep-learning-based baselines, including CNN, Bi-LSTM and Bi-LSTM+Attention, outperform BPR up to 6.9% on *Toys* dataset, which suggests that deep models are effective in capturing the behavior patterns and dependencies on different activities. However, although the deep-learning-based approach is slightly better than the ATRank on AUC[6], its training time is very long and cannot be compared with the efficiency of ATRank. (3) Bi-LSTM+Attention approach performs the best in all baselines because Bi-LSTM structure uses information from the past and the future, and captures the time signals. Moreover, attention mechanism models the correlations between different behaviors, which can better extract the specific patterns in sequences. (4) CSAN without self-attention and position encoding is comparable with BPR. Neither of these methods can model the sequence signals in the system. This indicates the importance of modeling dynamic contextual dependency by encoding temporal order. (5) Our proposed CSAN performs better than all the other baselines on five sub-datasets. The experiments show that our model is effective and valuable.

4.5.2 Zhihu. The experimental results of all the methods for Zhihu dataset are illustrated in Table 3 and Fig. 5. We have following observations: (1) All methods perform better on Zhihu dataset than on Amazon dataset because Zhihu dataset contains much richer information including multi-type actions and multi-modal content, which greatly improves the performance of the models. (2) Both AUC and precision@K metric demonstrate that CSAN achieves the best performance compared with all the baseline methods. We attribute the superiority of CSAN to its three properties: 1) Heterogeneous behavior embedding network is built to map all the multi-type and multi-modal behaviors into a common semantic space, to integrate heterogeneous content and extract useful information effectively. 2) Our self-attention mechanism works on feature-level instead of on element-level, which can select the features that can best describe the behavior's specific meaning in any given contextual scenario, and include this information in the sentence encoding output. 3) Bi-directional position encoding can capture long-term dependency and short-term interest better for the more accurate personalized recommendation.

[6]We experiment on the code released by ATRank without modifying any parameters, and the results are as shown in the table.

4.6 Model Component Analysis

There are three key differences between our method and the previous attention-based recommendation methods: the multi-type actions and multi-modal content, the feature-wise self-attention probability of context correlation, the position encoding matrices for dynamic dependency modeling. In this section, we remove these key components separately from our unified sequence modeling network, to examine the contributions of the proposed components in the model. As shown in Fig. 6, the AUC and precision will decrease in varying degrees without any part. Among them, the result of *without position encoding matrices* decreases the most which indicates the importance of position encoding to sequence recommendation. The decline of *without attention probablity* shows that our CSAN can effectively model the complicated contextual correlations. *Without image* and *action* shows that the necessity of modeling heterogeneous behavior.

4.7 Case Study

To see what contextual dependencies can be captured by our CSAN model, we generate the self-attention score heatmap of two Zhihu user behavior sequences in forward and backward respectively, as shown in the Fig. 7. We demonstrate the dependency at feature-level to observe the contextual information captured by different feature layers. Because the features are too much to be displayed one by one, we chose two representative layers. We select two behavior sequences from Zhihu data as examples. One is *{Parenting, Child rearing, Hollywood movies, Education, Music Education, Actress, Photography, Instructor, Education}*, the other is *{Monster hunter (Game), Hip-hop, Hollywood, Music production, Music Education, Music, American drama, Jazz, Europe music.}*. Both the two users have the same behavior *Music Education*. Note that we use each topic corresponding to the item which the user acts on as the representation of this behavior.

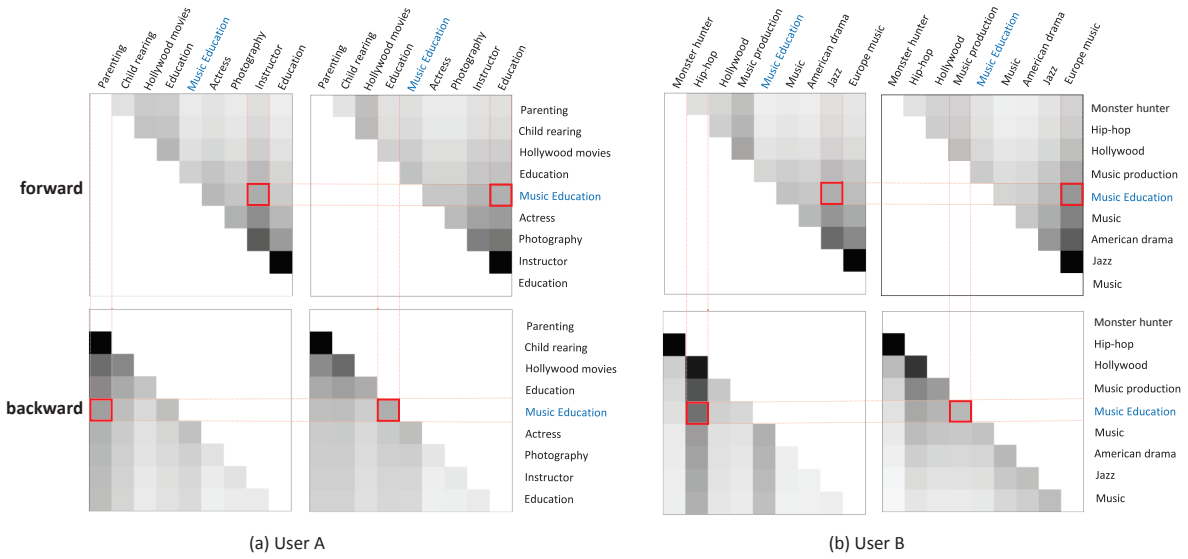

Figure 7: Forward and backward self-attention probabilities for two user behavioral sequences.

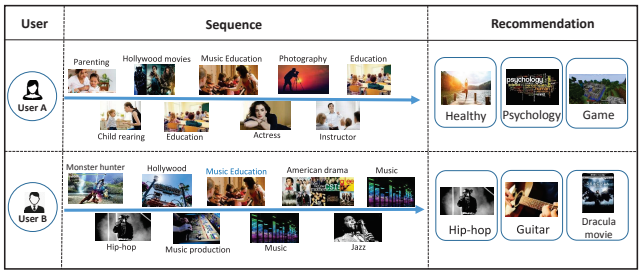

Figure 8: Recommendations made for two users by CSAN on Zhihu data.

Take behavior *Music Education* as an example, from the Fig. 7, we have several observations: (1) Different feature layers may correspond to different behaviors. For instance, as shown in the forward attention score of user A, the two feature layers focus on *Instructor* and *Education*. (2) Directional information helps to generate context behavior representations with position encoding. For example, the forward and backward processes capture *Instructor, Education* and *parenting, education*, respectively. Past and future information are taken into account to assist in context modeling. (3) The same behavior has different meanings in different context. For instance, for the same item *Music Education* the two users both act on, user A focuses on education-related topics, such as *Instructor* and *Parenting*, since we can roughly infer that user A may be an educator from his/her behavior sequence. As for user B, music-related themes are more closely related to the behavior because the sequence shows that user B may be a singer or a music lover. This demonstrates that our approach has the ability to model the polysemy of behaviors.

In Fig. 8, we display some recommendations made by CSAN on Zhihu data. Behavior sequences are fed into CSAN and the top-3 recommendations are generated. As we can see from these examples, CSAN can capture long-term dependency successfully. For instance, *Psychology* is recommended to user A who appears to be an educator, and *Guitar* to user B who seems to be singer or music lover. CSAN also captures short-term interest. For example, it recommends a dracula movie to user B since the user recently watched the American drama.

5 CONCLUSIONS

We introduce a contextual self-attention network, CSAN, for modeling the sequential behaviors in recommendation tasks. CSAN is a unified framework which can model with multi-type actions and multi-modal content based solely on attention mechanism. CSAN performs a feature-wise selection over the input sequence for a specific context to produce the context-aware representations. The position encoding matrices takes temporal order into account to model dynamic contextual dependency. We analyze the proposed model on both single-type behaviors datasets (Amazon) and multi-type multi-modal behaviors dataset (Zhihu). The experiment results show that CSAN achieves promising performances over the existing highly optimized individual models, and demonstrates its suitability for modeling complex behavior patterns.

In addition, we introduce a large-scale dataset for the sequential recommendation from Zhihu, which contains detailed information about tens of thousands of users' activities over the past year. In the future, we will explore the CSAN to use the proposed attention mechanism for more complicated tasks.

6 ACKNOWLEDGEMENT

This work was supported in part by the National Key Research and Development Program of China (No. 2017YFB1002804), the National Natural Science Foundation of China under Grants 61432019, 61720106006, 61572503 and 61702509, the Key Research Program of Frontier Sciences, CAS, Grant NO. QYZDJ-SSW-JSC039, the Open Projects Program of National Laboratory of Pattern Recognition, and the Beijing Municipal Science & Technology Commission (No. Z181100008918012), and the K.C.Wong Education Foundation.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR* abs/1409.0473 (2014).

[2] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where You Like to Go Next: Successive Point-of-Interest Recommendation.. In *IJCAI*, Vol. 13. 2605–2611.

[3] Szu-Yu Chou, Yi-Hsuan Yang, Jyh-Shing Roger Jang, and Yu-Ching Lin. 2016. Addressing cold start for next-song recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 115–118.

[4] Junyoung Chung, ÃĞaglar GülÃğehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* abs/1412.3555 (2014).

[5] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-Attention Neural Networks for Reading Comprehension. In *ACL*.

[6] Yong Du, Wei Wang, and Liang Wang. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1110–1118.

[7] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized Ranking Metric Embedding for Next New POI Recommendation.. In *IJCAI*. 2069–2075.

[8] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *ICML*.

[9] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 161–169.

[10] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 191–200.

[11] Liang Hu, Longbing Cao, Jian Cao, Zhiping Gu, Guandong Xu, and Dingyu Yang. 2016. Learning Informative Priors from Heterogeneous Domains to Improve Recommendation in Cold-Start User Domains. *ACM Transactions on Information Systems (TOIS)* 35, 2 (2016), 13.

[12] Minghao Hu, Yuxing Peng, and Xipeng Qiu. 2017. Reinforced mnemonic reader for machine comprehension. *CoRR, abs/1705.02798* (2017).

[13] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1725–1732.

[14] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[16] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. 1188–1196.

[17] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. *CoRR* abs/1703.03130 (2017).

[18] Duen-Ren Liu, Chin-Hui Lai, and Wang-Jung Lee. 2009. A hybrid of sequential rules and collaborative filtering for product recommendation. *Information Sciences* 179, 20 (2009), 3505–3519.

[19] Qiang Liu, Shu Wu, and Liang Wang. 2017. Multi-behavioral sequential prediction with recurrent log-bilinear model. *IEEE Transactions on Knowledge and Data Engineering* 29, 6 (2017), 1254–1267.

[20] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts.. In *AAAI*. 194–200.

[21] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 43–52.

[22] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

[23] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 5528–5531.

[24] Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukar Burget, and Jan Cernocky. 2011. Rnnlm-recurrent neural network language modeling toolkit. In *Proc. of the 2011 ASRU Workshop*. 196–201.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[26] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 811–820.

[27] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16126

[28] Huan Song, Deepta Rajan, Jayaraman J. Thiagarajan, and Andreas Spanias. 2018. Attend and Diagnose: Clinical Time Series Analysis Using Attention Models. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16325

[29] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. (2018).

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 6000–6010.

[31] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. ACM, 403–412.

[32] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-based Transactional Context Embedding for Next-Item Recommendation. (2018).

[33] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2, 1-3 (1987), 37–52.

[34] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*. 2048–2057.

[35] Shuangfei Zhai, Keng-hao Chang, Ruofei Zhang, and Zhongfei Mark Zhang. 2016. Deepintent: Learning attentions for online advertising with recurrent neural networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1295–1304.

[36] 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 *AAAI*, Vol. 14. 1369–1375.

[37] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 425–434.

[38] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. 2018. ATRank: An Attention-Based User Behavior Modeling Framework for Recommendation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16216

[39] Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. 1059–1068. https://doi.org/10.1145/3219819.3219823