Domain Switch-Aware Holistic Recurrent Neural Network for Modeling Multi-Domain User Behavior

Donghyun Kim*
Yahoo! Research
Sunnyvale, CA, United States
cartopy@gmail.com

Sungchul Kim
Adobe Research
San Jose, CA, United States
sukim@adobe.com

Handong Zhao
Adobe Research
San Jose, CA, United States
hazhao@adobe.com

Sheng Li
University of Georgia
Athens, GA, United States
sheng.li@uga.edu

Ryan A. Rossi
Adobe Research
San Jose, CA, United States
rrossi@adobe.com

Eunyee Koh
Adobe Research
San Jose, CA, United States
eunyee@adobe.com

ABSTRACT

Understanding user behavior and predicting future behavior on the web is critical for providing seamless user experiences as well as increasing revenue of service providers. Recently, thanks to the remarkable success of recurrent neural networks (RNNs), it has been widely used for modeling sequences of user behaviors. However, although sequential behaviors appear across multiple domains in practice, existing RNN-based approaches still focus on the single-domain scenario assuming that sequential behaviors come from only a single domain. Hence, in order to analyze sequential behaviors across multiple domains, they require to separately train multiple RNN models, which fails to jointly model the interplay among sequential behaviors across multiple domains. Consequently, they often suffer from lack of information within each domain. In this paper, we first introduce a practical but overlooked phenomenon in sequential behaviors across multiple domains, i.e., *domain switch* where two successive behaviors belong to different domains. Then, we propose a *Domain Switch-Aware Holistic Recurrent Neural Network* (DS-HRNN) that effectively shares the knowledge extracted from multiple domains by systematically handling *domain switch* for the multi-domain scenario. DS-HRNN jointly models the multi-domain sequential behaviors and accurately predicts the future behaviors in each domain with only a single RNN model. Our extensive evaluations on two real-world datasets demonstrate that DS-HRNN outperforms existing RNN-based approaches and non-sequential baselines with significant improvements by up to 14.93% in terms of recall of the future behavior prediction.

CCS CONCEPTS

• **Information systems → Web log analysis**; *Web searching and information discovery*;

*This work was done during his internship at Adobe Research.

WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3291019

KEYWORDS

Multi-domain user behavior, Sequence modeling, Domain switch, Recurrent neural network

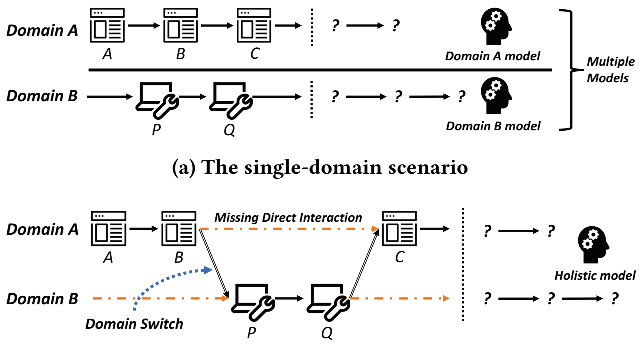
ACM Reference Format:
Donghyun Kim, Sungchul Kim, Handong Zhao, Sheng Li, Ryan A. Rossi, and Eunyee Koh. 2019. Domain Switch-Aware Holistic Recurrent Neural Network for Modeling Multi-Domain User Behavior. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3291019

1 INTRODUCTION

With the diversity of web services, today's users on the web frequently switch from one service to another within short period of time, which makes their sequential behavior patterns more complex and diverse. For example, users access to Youtube to watch videos, use then Google to search information related to the content of videos and re-access to Youtube to browse videos depending on their search results in Google even within a short amount of time. Typically, such sequential behavior across multiple domains[1] are locally and globally correlated. In this paper, we introduce it as *domain switches*[2] where two successive behaviors belong to different domains, e.g., from Youtube to Google and from Google to Youtube. Accordingly, from the marketer's point of view, accurately understanding *domain switches* from browsing web pages to clicking/buying items within each domain is an important key when deciding marketing strategies for each domain.

Nowadays, RNNs have been actively used to analyze sequential behaviors in the user behavior modeling [1, 12, 27] or recommendation [2–4, 16, 20] fields. They have demonstrated their superiority to traditional approaches [17, 18] by effectively considering sequential relationships of user behaviors [15]. However, to the best of our knowledge, existing RNN-based approaches have mostly focused on only a singe domain scenario of sequential behaviors as illustrated in Figure 1a. As a result, they require multiple independent RNN models for domains in order to analyze and predict domain-wise sequential behaviors for the multi-domain scenario. Unavoidably,

[1]In this paper, *domains* refer to service categories in a large-scale web service [24, 25].
[2]In our two real-world datasets. *domain switches* frequently occur with the average ratios, 0.29 and 0.32, of the number of domain switches to the length of sequences.



(a) The single-domain scenario

(b) The multi-domain scenario

Figure 1: Two illustrations of (a) the single-domain scenario with two behavior sequences for a user within *Domain A* or *B* and (b) the multi-domain scenario with an aggregated behavior sequence for the user across two domains.

they cannot exploit global dynamics of sequential behaviors contained across domains, which leads to an inferior performance of RNN models.

In this paper, we present a single RNN-based framework that handles the multi-domain scenario to accurately predict domain-wise future behavior. Precisely, the framework aggregates sequential behaviors of each user across multiple domains in chronological order. For example, the two sequences came from a single user in Figure 1a are merged into a sequence in chronological order as in Figure 1b. It naturally reveals the user's global behavior pattern across domains via *domain switches*. Moreover, even if some users have rarely behaved in a certain domain, we can expect that knowledge from other domains will complement insufficient information in analyzing their future behavior within the domain, which additionally mitigates *the cold-start problem*.

However, although the chronologically ordered aggregation preserves the global dynamics of sequential behaviors, some direct connections between two behaviors within each domain (i.e., local dynamics) are lost owing to domain switches in the multi-domain sequential behaviors of users. In Figure 1, when two sequences are aggregated in chronological order for modeling multi-domain sequential behaviors, pages B and C in *Domain A* are connected to pages P and Q in *Domain B*, respectively. With the domain switches, the direct interaction between B and C pages in *Domain A* are eventually lost, which results in information loss within the *Domain A*. This information loss aggravates as the number of the domain switches increases, which eventually hinders performance improvements for next behavior prediction within each domain.

To tackle this issue, we propose *Domain Switch-Aware Holistic Recurrent Neural Network* (DS-HRNN) that effectively addresses missing direct interactions in local dynamics on top of the RNN-based framework. Specifically, we first recover missing direct interactions caused by domain switches, and compute *domain switch-aware supplementary loss* with respect to missing direct interactions. Moreover, we reflect correlations between global and local past behaviors at the end of each domain switch by introducing *domain switch-aware behavior regularizer*. It is worth noting that in Figure 1b the past behaviors of page C are locally page B and globally page Q , which means that it is likely that a correlation between pages B and Q exists. These two techniques attentively take into account local dynamics at every domain switches. Consequently, they enable DS-HRNN not only to effectively reflect global dynamics into a single RNN model but also to preserve local dynamics without compromising further improvement in analyzing future behavior sequence within each domain. We conduct extensive experiments on two real-world datasets, and our experimental results show the effectiveness of DS-HRNN in terms of predicting next behavior sequence in each domain compared with existing approaches.

The main contributions of this work are as follows:

- We address the problem of modeling user sequential behavior across multiple domains and propose a single RNN-based framework to fully leverage multi-domain user behavior.
- We introduce the *domain switch* phenomena and propose DS-HRNN on top of the RNN-based framework by devising two domain switch-aware techniques to boost its predictability.
- We show that DS-HRNN outperforms the state-of-the-arts, especially under the cold-start evaluation much better. Also, we investigate the impact of two proposed techniques together with a case study.

2 RELATED WORK

2.1 RNN-based Sequential Behavior Modeling

In order to effectively model sequential behaviors, a RNN-based recommeder system was firstly introduced for session-based recommendation [6]. On top of that, a variety of RNN-based approaches for the next behavior prediction have been developed by additionally considering personalization [4, 16, 22], context-awareness [3, 19, 27], and different types of user behavior [1, 12, 21, 26].

Personalization. Donkers et al. [4] devised a user-based Gated Recurrent Units (GRU) that attentively consider user embeddings along with sequential item information for personalized next item recommendations. Quadrana et al. [16] hierarchically exploited a user-level and a session-level RNNs to reflect users' inter-session sequential dynamics into intra-session sequential dynamics for personalized session-based recommendations. Wu et al. [22] exploited a user-level and an item-level RNNs in parallel based on the user-level and the item-level history, respectively, to consider temporal evaluation of users and items for the rating prediction.

Context-awareness information. Beutel et al. [3] and Smirnova et al. [19] reflected contextual information into input, output and RNN layers. Particularly, they parametrized hidden state transitions in RNNs with an element-wise multiplicative function of context embeddings for better next item recommendations and compared their approach with baselines without contexts. Zhu et al. [27] claimed the importance of taking into account time intervals in order to effectively capture the relations of user behaviors, and thus, they devised a new LSTM variant to equip LSTM [7] with newly introduced time gates to model time intervals between two successive user behaviors for the next-basket recommendations.

Different types of user behavior. Zhou et al. [26] introduced micro-behaviors such as click source, browsing modules, and cart and order based on users' sequential behavior in the e-commerce. They simultaneously exploited user behavior as well as the corresponding micro-behavior as an input of RNN, and used attention mechanism [14] on top of outputs of RNN for better next item recommendation. Similar to Zhou et al. [26], Twardowski [21] exploited event types of behaviors (e.g., search, view and watch/cart) together with users' behavior history for enhancing session-based recommendation without the attention mechanism. Liu et al. [12] integrated the log-bilinear model [14] into RNN so that hidden sates can capture short-term and long-term contexts in user behavior sequences for predicting what a user will choose next. In addition to that, they tried to model multiple types of behaviors with behavior-specific transition matrices in their model.

However, none of existing RNN-based approaches have tried to consider the multi-domain scenario as in Figure 1b. Specifically, even if different types of behavior that are used as additional information of inputs are assumed to be domains, their approaches inevitably fall into the single-domain scenario since their objectives are modeled to predict next behaviors of users regardless of types of behaviors. Moreover, in existing approaches, one behavior can be concurrently assigned to multiple types such as click, add-to-cart and purchase, however, the multi-domain scenario has the distinctive property where one behavior is exactly assigned to only one domain. Therefore, how to consider the multi-domain scenario in users' sequential behaviors still remains a challenging problem.

2.2 Multi-Domain User Behavior Modeling

As one of cross-domain approaches, multi-domain user behavior modeling approaches have been explored mainly based on non-sequential methods such as matrix factorization [8, 23–25] and feed forward neural network [5]. It is worth noting that these approaches seamlessly learn shared knowledge *across all domains* so that the shared knowledge can be effectively used for better domain-wise user behavior prediction. Particularly, Li et al. [8] and Zhang et al. [25] utilized a cluster-level rating matrix from multiple rating matrices in order to share the knowledge collected from multiple domains. Zhang et al. [24] simultaneously dealt with multiple matrix factorization tasks in different domains while modeling the correlations between domains through covariance matrix obtained from multiple user latent models for domains. Elkahky et al. [5] proposed a single multi-view deep neural network model that jointly learns dense features of items from different domains such as News, Apps and Movie/TV via common users. Yang et al. [23] observed that users have cross-site as well as site-specific preferences in multiple video websites, and thus they proposed a matrix factorization based model that infers site-specific user variables based on cross-site user variables. In their model, these site-specific user variables are used to predict ratings with video variables.

However, these approaches do not seamlessly take into account sequential dynamics of user behaviors. To the best of our knowledge, our work is the first attempt to consider sequential dynamics as well as the multi-domain scenario in order to effectively predict future behaviors for each domain. Note that multi-task learning-based RNNs [9–11, 13] in the natural language processing (NLP) field might be regarded as our related work, but they differ from ours with the following reasons. With respect to objectives, existing approaches for NLP mainly try to simultaneously solve multiple classification problems based on text whereas our approach try to predict domain-wise next behavior based on previous behavior history across multi-domain. Moreover, inputs of existing approaches come from only a single source (e.g., English) whereas those of our approach come from multiple sources (e.g., domain A, B and C). For example, two pairs of a sentence and a label from different tasks can share English words, however, sequences of behaviors from different domains never have common behaviors. Thus, the task of modeling multi-domain user behavior is more challenging than that of multi-task learning in NLP. For clarity of exposition, we formally define our problem in the following section.

3 METHODOLOGY

3.1 Problem Statement

In this paper, we assume that multi-domain user behaviors are browsing logs from different domains on a large-scale web service. Let $\mathcal{B}^{(d)}$ denote a set of behaviors in domain $d \in \mathcal{D}$ where $\mathcal{D}$ denotes a finite set of domains on the service. We assume that behavior sets do not have intersection[3], i.e., $\bigcap_d \mathcal{B}^{(d)} = \emptyset$ since different service categories are highly likely not to share their entries such as pages and products. With this setting, the task is to accurately predict domain-wise next behavior sequences $\mathcal{S}_i = \{S_i^{(1)}, S_i^{(2)}, \cdots, S_i^{(d)}\}$ with given behavior histories over all domains $\mathcal{X}_i = \{X_i^{(1)}, X_i^{(2)}, \cdots, X_i^{(d)}\}$, where $S_i^{(d)} = \{y_{i,t} | y_{i,t} \in \mathcal{B}^{(d)}, t > T\}$ denotes a sequence of future behaviors for user i in domain d and $y_{i,t}$ denotes a future behavior for user i in domain d at time t. Similar to $S_i^{(d)}$, $X_i^{(d)} = \{x_{i,t} | x_{i,t} \in \mathcal{B}^{(d)}, 1 \le t \le T\}$ denotes a user behavior history in a domain where $x_{i,t}$ denotes a past behavior for user i in domain d at time t. Note that a user can be replaced with a session since multi-domain behaviors can exist even within a session. In this paper, a user and a session are interchangeable.

3.2 RNN-based Framework for Multi-Domain

According to the task, since we consider sequential dynamics of behaviors, the training objective function of RNN-based approaches for each user[4] can be commonly represented as:

$$\mathcal{L}(\theta) = \sum_d \sum_{t=2} l(x_t, \mathcal{F}_\theta(\mathbf{x}_{<t})), \tag{1}$$

where l is a loss function such as cross entropy, and $x_t \in \mathcal{B}^{(d)}$ denotes a true behavior in a current target domain d at time t. $\mathcal{F}_\theta$ is the function based on RNNs, and $\mathbf{x}_{<t}$ denotes sequential behaviors before t time in chronological order. Typically, RNNs have a recurrent form such that the hidden state $h_{t-1} \in \mathbb{R}^k$ at time step $t-1$ is computed recursively from the previous hidden state and the current input:

$$h_{t-1} = f(x_{t-1}, h_{t-2}),$$

where f denotes a RNN cell such as the basic RNN, LSTM, and GRU cell, and in our task, h_{t-1} is used to estimate x_t. It is worth noting that $\mathcal{F}_\theta$ and $\mathbf{x}_{<t}$ vary depending on how RNN-based approaches consider sequential dynamics as well as multi-domain user behaviors. In this section, we firstly describe how existing approaches for

[3] This assumption is more challenging according to [8].

[4] We omit user indices since all users are handled in the same way.

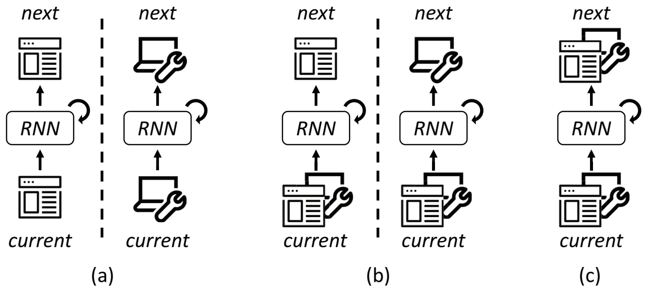


Figure 2: Illustrations about RNN-based approaches of (a) the SM case, (b) the MM case and (c) the MS case.

single domain scenario [1, 3, 4, 6, 12, 16, 19, 21, 22, 26, 27] can be simply applied to formulate $\mathcal{F}_\theta$ together with $\mathbf{x}_{<t}$ for our multi-domain scenario. Then, we present two naive RNN-based approaches to formulate them for our multi-domain scenario.

Single-domain based approach with Multiple models (SM). Related to existing RNN-based approaches, Figure 2a illustrates how the single-domain based approach works for modeling multi-domain user sequential behaviors. Intuitively, this approach necessarily requires multiple RNN models for domains since it independently handles sequences from different domains. To be precise, $\mathcal{F}_\theta$ and $\mathbf{x}_{<t}$ in this approach are defined as:

$$\left.\begin{array}{rcl} \mathcal{F}_\theta & = & \mathcal{F}_{\theta^{(d)}} \\ \mathbf{x}_{<t} & = & \mathbf{x}_{<t}^{(d)} \end{array}\right\} \quad \text{if } x_t \in \mathcal{B}^{(d)}$$

where $\mathcal{F}_{\theta^{(d)}} \in \{\mathcal{F}_{\theta^{(1)}}, \mathcal{F}_{\theta^{(2)}}, \ldots\}$ is a RNN model parameterized by $\theta^{(d)}$ only for domain d, and $\mathbf{x}_{<t}^{(d)}$ consists of behaviors only within domain d as a subset[5] of the entire sequential behaviors $X^{(d)}$ for a user in domain d. With given user behavior history $\mathbf{x}_{<t}^{(d)}$, each model $\mathcal{F}_{\theta^{(d)}}$ computes the probability distribution $\mathbf{p}_t^{(d)}$ of next behaviors in the corresponding domain as following:

$$\begin{aligned} \mathcal{F}_{\theta^{(d)}}(\mathbf{x}_{<t}) &= \mathbf{p}_t^{(d)} \\ &= \frac{1}{Z^{(d)}}[p_{\theta^{(d)}}(x_t = b|\mathbf{x}_{<t}^{(d)})]_{\forall b \in \mathcal{B}^{(d)}} \end{aligned}$$

where $p_{\theta^{(d)}}$ indicates the predicted probability for the next user behavior b within domain d, and $Z^{(d)}$ is a normalizing factor for domain d. However, since this approach independently trains $\theta^{(d)}$ with $\mathbf{x}_{<t}^{(d)}$ based on an actual target x_t for all domains, knowledge from different domains cannot be shared to figure out global dynamics of sequential behaviors. Furthermore, this isolated knowledge without global dynamics cannot complement lack of information each other.

Multi-domain based approach with Multiple models (MM). To address the issue of **SM** approach, Figure 2b shows our first RNN-based naive approach. Particularly, we aggregate user behaviors over all domains in chronological order, and consider them as inputs of RNN models to reflect global dynamics of sequential behaviors into RNN models. Accordingly, we redefine $\mathbf{x}_{<t}$ as:

$$\mathbf{x}_{<t} = \bigcup_d \mathbf{x}_{<t}^{(d)}$$

[5] If t is the length of a sequence plus one, $\mathbf{x}_{<t}^{(d)}$ becomes $X^{(d)}$.

Then, $\mathbf{x}_{<t}$ stands for chronologically aggregated user behaviors across multiple domains, and $\mathbf{p}_t^{(d)}$ with $\mathbf{x}_{<t}$ has a new form of:

$$\mathbf{p}_t^{(d)} = \frac{1}{Z^{(d)}}[p_{\theta^{(d)}}(x_t = b|\mathbf{x}_{<t})]_{\forall b \in \mathcal{B}^{(d)}}$$

From the above equation, in **MM** approach, behaviors from non-target domains are additionally exploited to infer next behaviors for a target domain. This enables not only to reflect global dynamics of user sequential behaviors across multiple domains but also to complement lack of behavioral information, especially for users who rarely behave in the target domain. However, this approach still requires as many training procedures as the number of RNN models to train each $\theta^{(d)}$. Moreover, behaviors from non-target domains do not fully help to tackle the issue of **SM** approach for the target domain, because each RNN model is individually trained by considering only the loss of its target domain while neglecting the loss of non-target domains.

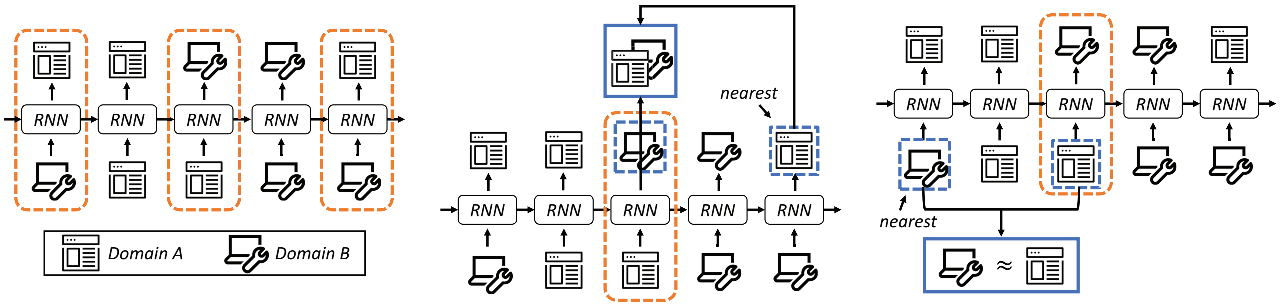
Multi-domain based approach with a Single model (MS). To handle the issue of **MM** approach where each RNN model will be solely optimized for the target domain, we additionally reflect loss of non-target domains into each RNN model of the **MM** approach which is previously trained for its target domain. Then, it enables each RNN model to fully leverage behaviors across multi-domain, and eventually all RNN models work exactly the same since the loss of a non-target domain in a RNN model is regarded as the loss of a target domain in other RNN model. Accordingly, $\forall\theta^{(d)}$ can be replaced with a global model parameter θ^g for a single RNN model across multi-domain. With θ^g, $\mathcal{F}_{\theta^{(d)}}$ becomes $\mathcal{F}_{\theta^g}$, and $\mathbf{p}_t^{(d)}$ has the final form of:

$$\mathbf{p}_t^{(d)} = \frac{1}{Z^{(d)}}[p_{\theta^g}(x_t = b|\mathbf{x}_{<t})]_{\forall b \in \mathcal{B}^{(d)}}$$

Figure 2c illustrates the **MS** approach that exploits a single RNN model to fully consider chronologically aggregated user behaviors across multiple domains. Note that the **MS** approach is totally different from simple RNN-based approaches based on the assumption where user behaviors do not belong to any domains, because the **MS** approach computes the domain-wise loss and prediction. If the loss and prediction are globally computed, predicting sequences of the next user behaviors tends to be inaccurate, because all behaviors are considered as candidates for the next behaviors regardless of domains.

3.3 Domain Switch-Aware Holistic Recurrent Neural Network (DS-HRNN)

The key idea of handling multi-domain user behavior are 1) aggregating user behaviors from multiple domains into one sequence in chronological order and 2) exploiting a single RNN model that takes sequences of multi-domain user behaviors. However, in the **MS** approach for the single RNN model, *domain switches* by the chronologically ordered aggregation are highly likely to hinder further improvements in multi-domain behavior prediction. The reason is that considering the loss of all domains in the single RNN model gives rise to the disconnection of direct interactions between behaviors in the same domain (i.e., local dynamics). To better understand the issue of the **MS** approach, Figure 3a shows



(a) Example of domain switches (b) Supplementary loss (c) Behavior regularizer

Figure 3: Example of (a) domain switches and our proposed two strategies for DS-HRNN – Domain switch-aware (b) supplementary loss and (c) behavior regularizer. Dotted orange line denotes domain switches.

the unfolded structure of the single RNN model with the cases of domain switches. From the view point of *Domain A*, the second domain switch causes the RNN model to be trained for predicting different behaviors out of *Domain A* even though the current input belongs to *Domain A*. For *Domain B*, the first and last domain switch impair the RNN model in a similar way in which the second does. Eventually, the ill-trained single RNN model results in inaccurate behavior prediction for each domain. Here, we attribute this issue to the broken local dynamics, and we introduce our novel RNN-based approach, *Domain Change-Aware Holistic Recurrent Neural Network* (DC-HRNN) by proposing two types of domain switch-aware techniques to alleviate the disconnection of local dynamics. Then, we formulate the final objective function and briefly describe the architecture of the single RNN model.

Domain switch-aware supplementary loss. The key idea for alleviating the disconnection of local dynamics is to recover the disconnection as much as possible while preserving global dynamics of sequential behaviors simultaneously. To this end, in terms of outputs, we define a *domain switch-aware supplementary loss* that is an explicit way of recovering the lost connection at domain switches $\mathcal{L}_s$ as follows.

$$\mathcal{L}_s(\theta^g) = \sum_{t=2} \mathbb{I}[x_{t-1} \in \mathcal{B}^{(d)} \wedge x_t \notin \mathcal{B}^{(d)}]\, l(x_{t-1+n^{(d)}}, \mathcal{F}_{\theta^g}(\mathbf{x}_{<t}))$$

where $\mathbb{I}[\cdot]$ is the indicator function, and $n^{(d)}$ is the distance from the current input $x_{t-1} \in \mathcal{B}^{(d)}$ to the nearest next behavior in the same domain d. Note that the indicator function lets the supplementary loss valid only at domain switches since disconnection does not occurs on other than domain change interactions. Figure 3b illustrates how $\mathcal{L}_s$ works in the single RNN model. Precisely, $\mathcal{L}_s$ searches the nearest next behavior which belong to the same domain, *Domain A*, as the current input of the RNN model on the domain switch, and $\mathcal{L}_s$ computes the correlation between these two behaviors. Through $\mathcal{L}_s$, we aim to explicitly inject local dynamics previously lost in the *M-S* approach into the RNN model, which leads to more accurate sequential behavior prediction for each domain.

Domain switch-aware behavior regularizer. Similarly, alleviating the disconnection of local dynamics can be also achieved in an implicit way in terms of inputs. To be specific, Figure 3c shows that the input on the domain switch is likely to be correlated with the past nearest behavior who belongs to the same domain as the output on the domain switch. The reason is that these two behaviors for inputs can share the same output as the next behavior. Note that the past nearest behavior can be regarded as the past behavior of the current output in terms of local dynamics of the domain, *Domain B*. Eventually, local dynamics previously lost in the M-S approach can be implicitly recovered via taking into account these correlations for inputs. To reflect this, we introduce a domain switch-aware behavior regularizer $\mathcal{L}_r$ that minimizes the distance between correlated inputs from different domains:

$$\mathcal{L}_r(\theta^g) = \sum_{t=2} \mathbb{I}[x_{t-1} \notin \mathcal{B}^{(d)} \wedge x_t \in \mathcal{B}^{(d)}] \| \mathcal{F}_{\theta^g_{in}}(x_{t-m^{(d)}}) - \mathcal{F}_{\theta^g_{in}}(x_{t-1}) \|_2$$

where $\mathcal{F}_{\theta^g_{in}}(x) \in \mathbb{R}^k$ denotes the k-dimensional embedding vector for input behavior x, and $m^{(d)}$ is the distance from the current input x_{t-1} to the nearest past behavior which belongs to the same domain d as the output x_t. This regularizer might seem to play the same role as the supplementary loss, however, the regularization focuses on building correlated input embeddings for different domains of behaviors, which is not explicitly taken into account in the supplementary loss technique.

Final objective & Architecture. Given our two techniques on top of the **M-S** approach, our final objective $\mathcal{J}(\theta^g)$ for all users to be minimized is formulated as follows:

$$\mathcal{J}(\theta^g) = \sum_{user} (\mathcal{L}(\theta^g) + \lambda_s \cdot \mathcal{L}_s(\theta^g) + \lambda_r \cdot \mathcal{L}_r(\theta^g))$$

where λ_s and λ_r are loss coefficients for the domain switch-aware supplementary loss and the domain switch-aware behavior regularizer, respectively. For $\mathcal{L}$ and $\mathcal{L}_s$, we use the cross entropy loss to deal with the probability distribution of next behaviors. Our architecture for $\mathcal{F}_{\theta^g}$ consists of three parts: 1) *Input module*, which projects raw sparse input of each behavior (e.g., one-hot vector) into low dimensional embedding space, 2) *Recurrent module*, which recursively updates the hidden state from the previous hidden state and the current input in order to reflect sequential information, 3) *Output module*, which computes the probability distribution of next behaviors for each domain. Specifically, the process of $\mathcal{F}_{\theta^g}$ with a

Table 1: Data statistics of two datasets: uniq. and avg. denote unique behaviors and the average length of sequences.

Dataset	Dom.	# Behav. (# uniq.)	# Seq. (avg.)
CorWeb	*All*	4,858,663 (15,309)	85,048 (57.13)
	Domain A	1,577,354 (1,309)	-
	Domain B	686,226 (5,664)	-
	Domain C	2,595,083 (8,336)	-
Yoochoose	*All*	1,560,414 (20,197)	50,698 (30.78)
	Normal	783,722 (17,287)	-
	Special	776,692 (2,910)	-

given sequence $\mathbf{x}_{<t}$ and $x_t \in \mathcal{B}^{(d)}$ is as follows:

$$e_{t-1} = \mathcal{F}_{\theta^g_{in}}(x_{t-1}) = E \cdot x_{t-1} \qquad (Input)$$

$$h_{t-1} = \mathcal{F}_{\theta^g_{rec}}(e_{t-1}, h_{t-2}) = f(e_{t-1}, h_{t-2}) \qquad (Recurrent)$$

$$\mathbf{p}^{(d)}_t = \mathcal{F}_{\theta^g_{out}}(h_{t-1}) = \text{Softmax}(W^{(d)}_o h_{t-1} + b^{(d)}_o) \qquad (Output)$$

where $\theta^g = \{\theta^g_{in}, \theta^g_{rec}, \theta^g_{out}\}$ and $e_{t-1} \in \mathcal{R}^k$ denotes the embedding vector for the user behavior at time $t-1$. $x_{t-1} \in \mathbb{R}^{|\mathcal{B}|}$ is the one-hot vector for the user behavior in the sequence at time $t-1$, and $E \in \mathbb{R}^{k \times |\mathcal{B}|}$ is a behavior embedding matrix for inputs of all the user behaviors. As mentioned in Section 3.2, $f(\cdot)$ denotes a RNN cell, which can be easily replaced with more complex RNN cells [3, 7, 12, 19, 27] for further improvements. In the output module, $W^{(d)}_o \in \mathcal{R}^{|\mathcal{B}^{(d)}| \times (k)}$ and $b^{(d)}_o$ denote the transformation matrix and bias, respectively in order to compute probability distribution of next behaviors for each domain. From the final objective and this architecture, we compute the gradient for parameters in θ^g, and update them by using mini-batch stocastic gradient descent (SGD) with learning rate η as follows:

$$\theta^g \leftarrow \theta^g - \eta \times \frac{\partial \mathcal{J}(\theta^g)}{\partial \theta^g}$$

After training, the next behavior prediction is conducted with $\mathcal{F}_{\theta^g}$ and the last behavior x_T for a user in the training set:

$$\begin{aligned} \mathbf{p}_{T+1} &= \mathcal{F}_{\theta^g}(x_T, \mathbf{x}_{<T}) \\ y_{T+1} &= \arg\max_b [\mathbf{p}^{(d)}_{T+1}]_{1 \le b \le |\mathcal{B}^{(d)}|} \end{aligned}$$

This prediction process is repeated via previous prediction results to generate a sequence $S^{(d)} = \{y_t | y_t \in \mathcal{B}^{(d)}, t > T\}$ of future behaviors in the domain d for a user.

4 EXPERIMENTS

In this section, we evaluate the empirical performance of DS-HRNN on real-world datasets, and thus we design experiments to verify the following research questions (RQs):

RQ 1 Does DS-HRNN outperform other competitors?
RQ 2 By considering multi-domain user bahavior, does DS-HRNN indeed mitigate to the cold-start problem?
RQ 3 How do the supplementary loss and behavior regularizer affect the model performance?

4.1 Experimental Setup

Dataset. In order to validate our approaches, we employ two real-world datasets of user browsing logs on the web: a Corporate Website (CorWeb) and Yoochoose[6]. The CorWeb dataset consists of web browsing logs of users on a corporate website, which contains tuples of a hashed user id, page URL, and timestamp. We regard visited URLs as user behaviors, and annotate them with the domain tags based on their base domain so that each behavior belongs to only one domain among the *Domain A*, *B*, and *C*. The Yoochoose dataset consists of item browsing logs of users on the e-commerce site, Youchoose, which contains tuples of session id, item id, timestamp and category. We regard clicked items as user behaviors, and annotate them with the domain tags so that they are categorized into two disjoint domains; *Normal* and *Special* based on category information whose items belong to the 'special offer' category at least one or not. For both datasets, each sequence is comprised of multi-domain behaviors in chronological order through common user ids and session ids, respectively. We preprocess data so that the length of each sequence is at least 10, and Table 1 shows the statistics of both datasets.

Competitors. We evaluate DS-HRNN with baselines categorized into single domain-based approaches (e.g., **SM**) and multiple domain-based approaches (e.g., **MM**, **MS**) as follows.

Single domain-based approaches.

- **BoB**: A latent model-based approach, which considers current behaviors as well as all previous behaviors as a bag of behaviors.
- **Covi**: A latent model-based approach, which considers the interaction of co-browsing two behaviors.
- **RNN$^{\text{SM}}$**: The RNN-based approach for a single domain as discussed in Section 3.2. Since RNNs have already surpassed other single domain-based approaches such as KNN [18] and MF [17], we omit them for brevity.

Multiple domain-based approaches.

- **MPF**: The state-of-the-art collective matrix factorization approach for multi-domain scenario, which considers domain-specific user latent models as well as shared user latent models across domains [23].
- **RNN$^{\text{MM}}$**: Our RNN-based approach with multiple models as in Section 3.2
- **RNN$^{\text{MS}}$**: Our RNN-based approach with a single model as in Section 3.2.

For fair comparisons among RNN-based approaches, we commonly uses the GRU cell for all the RNN-based approaches. Note that DS-HRNN can also exploit variants of RNN cells developed recently as mentioned in Section 3.3 and BoB and Covi were used as competitors of the single-domain based RNN approach [19].

Evaluation protocol. Since our task is to predict domain-wise next sequential behaviors from a certain time for user or session i, we used last five sequential behaviors $\mathcal{S}_i$ in each global sequence across multi-domain for testing, and the rest $\mathcal{X}_i$ are used for training. Note that $\mathcal{S}_i$ composes domain-wise true behavior sequences $S^{(d)}_i$ as many as the number of domains at most, and $\forall_d \mathcal{S}^{(d)}_i$ are never used as the inputs during testing. For tuning hyper-parameters and

[6]This dataset is publicly available at http://2015.recsyschallenge.com/challenge.html

Table 2: Overall performance for domain-wise next behavior predictions of various approaches. * denotes the second best. All the improvements of DS-HRNN over the second best (vs.Best) are statistically significant (All p values « .01)

			Single domain-based			*Mutiple domain-based*				*Improvements*	
Dataset	Domain	Measure	BoB	Covi	RNN^{SM}	MPF	RNN^{MM}	RNN^{MS}	DS-HRNN	vs.RNN^{SM}	vs.Best
CorWeb	Domain A	Recall@5	0.6212	0.5687	0.7508	0.4687	0.7873*	0.7832	**0.8010**	6.68%	1.73%
		Recall@10	0.7555	0.6393	0.8249	0.5746	0.8519*	0.8457	**0.8735**	5.88%	2.53%
		Recall@20	0.8486	0.7012	0.8795	0.6915	0.9087*	0.9046	**0.9257**	5.25%	1.87%
		MRR	0.4218	0.4009	0.5872	0.3850	0.6422*	0.6356	**0.6460**	10.02%	0.59%
	Domain B	Recall@5	0.2010	0.2317	0.2732	0.1114	0.2972*	0.2955	**0.3126**	14.43%	5.21%
		Recall@10	0.2850	0.3104	0.3599	0.1732	0.3925*	0.3919	**0.4136**	14.93%	5.37%
		Recall@20	0.3767	0.3969	0.4536	0.2390	0.4976*	0.4976	**0.5201**	14.68%	4.54%
		MRR	0.1431	0.1690	0.2017	0.0692	0.2177*	0.2166	**0.2272**	12.61%	4.36%
	Domain C	Recall@5	0.2943	0.3514	0.4161	0.1931	0.4204*	0.4137	**0.4386**	5.39%	4.33%
		Recall@10	0.4190	0.4572	0.5407*	0.2873	0.5279	0.5238	**0.5587**	3.33%	3.33%
		Recall@20	0.5571	0.5775	0.6584*	0.4166	0.6324	0.6363	**0.6727**	2.18%	2.18%
		MRR	0.1979	0.2485	0.2930	0.1429	0.3060*	0.3019	**0.3162**	7.93%	3.32%
Yoochoose	Normal	Recall@5	0.1177	0.1532	0.1717	0.0181	0.1802	0.1829*	**0.1888**	10.00%	3.27%
		Recall@10	0.1782	0.2200	0.2403	0.0311	0.2529	0.2562*	**0.2632**	9.54%	2.76%
		Recall@20	0.2488	0.2965	0.3158	0.0520	0.3324	0.3361*	**0.3425**	8.45%	1.90%
		MRR	0.0811	0.1077	0.1227	0.0160	0.1295	0.1312*	**0.1350**	10.02%	2.89%
	Special	Recall@5	0.1501	0.1942	0.2598	0.1310	0.2683*	0.2679	**0.2758**	6.15%	2.79%
		Recall@10	0.2288	0.2713	0.3434	0.2032	0.3525	0.3526*	**0.3623**	5.52%	2.76%
		Recall@20	0.3261	0.3544	0.4334	0.2905	0.4449*	0.4436	**0.4525**	4.40%	1.71%
		MRR	0.1052	0.1354	0.1925	0.0892	0.1988*	0.1984	**0.2037**	5.82%	2.46%

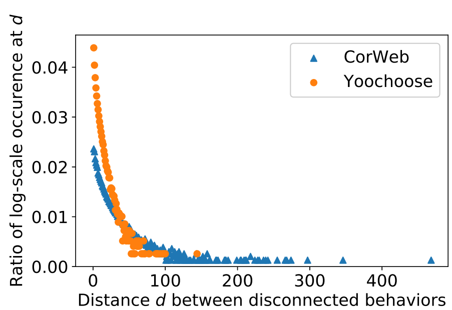


Figure 4: Statistics of the number of occurrences with distances between disconnected behaviors in the same domain

Table 3: Cold-start evaluation in terms of Recall@10. * denotes the second best.

Dataset	Domain	RNN^{SM}	RNN^{MM}	RNN^{MS}	DC-HRNN
CorWeb	Domain A	0.8060	0.8416*	0.8233	**0.8542**
	Domain B	0.3093	0.3662	0.3695*	**0.4001**
	Domain C	0.5052*	0.5027	0.4988	**0.5392**
Yoochoose	Normal	0.2242	0.2469	0.2561*	**0.2629**
	Special	0.3011	0.3214	0.3524*	**0.3631**

early stopping to avoid over-fitting, we user-wisely divide $\forall_i \mathcal{S}_i$ in half so that the first and second half are used for validating and testing, respectively. Based on the validation set, we tune all of the hyper-parameters by grid search with the dimension size of hidden states or latent models $k \in \{100, 300, 500\}$ and the loss coefficients $\lambda_s, \lambda_r \in \{0.0, 0.01, 0.1\}$. We commonly set the L_2 regularization coefficient to 1e-5, however, the coefficient for MPF is exceptionally set to 0.1, which gives better performance. We set the size of mini-batch to 256 and use Adam optimizer with an initial learning rate η of 0.001.

Evaluation metrics. We select two widely used metrics in RNN-based user behavior modeling approaches [4, 6, 16, 19, 26].

- **Recall@K**: The proportion of cases where a true behavior exist in top-K predicted behaviors at each future timestamp with $K \in \{5, 10, 20\}$.
- **MRR**: The average of the reciprocal ranks of a true behavior among all predicted behaviors at each future timestamp. We use this metric as a criterion during validation since this metric takes into account all behaviors. In the case of single model-based approaches, the average of MRRs across multi-domain is regarded as the criterion.

For reliability of our results, we conduct this evaluation for each domain five times with different initialization of trainable variables and different validation set, and we report the mean values in terms of our metrics.

4.2 Experimental Results

Overall performance evaluation (RQ 1). Table 2 shows the test performance on predicting next user behaviors within each domain in terms of Recall@K and MRR. For single domain-based approaches, we observe that the empirical performance is generally enhanced in the order of BoB, Covi and RNN^{SM}. As previously well known, this tendency shows that understanding sequential dynamics is significantly important for the next user behavior prediction. Even in multiple domain-based approaches, RNN-based approaches outperform MPF since MPF cannot consider sequential dynamics although MPF takes into account user latent models as well as multi-domain user behaviors. With respect to comparisons among RNN-based approaches, multi-domain based RNN significantly outperforms RNN^{SM}, which shows that understanding multi-domain user behaviors enables to enhance the next user behavior prediction

Table 4: Effects of loss coefficients for two proposed techniques on the CorWeb dataset (Recall@10)

λ_s \ λ_r	Domain A			Domain B			Domain C		
	0.0	0.01	0.1	0.0	0.01	0.1	0.0	0.01	0.1
0.0	0.8457	0.8455	0.8529	0.3919	0.3921	0.3965	0.5238	0.5243	0.5295
0.01	0.8575	0.8580	0.8642	0.3950	0.3964	0.4004	0.5345	0.5354	0.5394
0.1	0.8697	0.8699	**0.8735**	0.4100	0.4106	**0.4136**	0.5552	0.5560	**0.5587**

Table 5: Case study on two anonymized url sequences in the CorWeb – Left: *Last* is the last behavior of an aggregated user sequence in the training set. *Future* are the next true behaviors in the different domain and in the same domain as *Last*, respectively from the test set. Right: *rel.sim.* = $\cos(h_l, w_{same})/\cos(h_l, w_{diff})$ **where** h_l **is the hidden state with *Last* and** w **is from the trained output embedding for the next true behavior.** $\cos(A, B)$ **denotes a cosine similarity between A and B. *rank* is the position of the next true one in the ranked list of predicted ones.**

	Case 1	**Case 2**
Last	support/productX/error_update (*Domain B*)	support/productX/help/libraries (*Domain B*)
Future	others/productX/update (*Domain C*) support/productX_sub/tutorial (*Domain B*)	productX/authenticated (*Domain A*) support/productX/help/libraries (*Domain B*)

	Case 1		**Case 2**	
Model	*rel.sim.*	*rank*	*rel.sim.*	*rank*
RNN^{MS}	1.1002	22	0.8688	10
DS-HRNN	**1.4301**	**13**	**1.0601**	**2**

for each domain. We also observe that RNN^{MM} tends to slightly performs better than RNN^{MS} on the CorWeb dataset. This observation well demonstrates that the explicit disconnection of local dynamics in the **MS** approach caused by chronologically ordered aggregation hinder improvements in multi-domain user behavior prediction as mentioned in Section 3.3. However, in the Yoochoose dataset, we found that the performance of RNN^{MM} is comparable or worse than that of RNN^{MS}. To figure out this phenomenon, we investigate statistics of two datasets about the number of occurrences of distances between disconnected behaviors within the same domain as shown in Figure 4. Interestingly, most pairs of disconnected behaviors are closer in the Yoochoose dataset than in the CorWeb dataset. Based on this observation, we discover that the negative effects of the chronologically ordered aggregation in RNN^{MS} are highly related to distances between disconnected behaviors. Lastly, it is noteworthy that our final proposed approach, DS-HRNN, considerably beats all the competitors over all the cases, which demonstrates that multi-domain behaviors are beneficial and our proposed techniques are indeed effective. Particularly, compared with RNN^{SM}, DS-HRNN achieves significant improvements up to almost 15%. Note that *Domain B* has the smallest number of behaviors as in Table 1, and thus we regard that multi-domain behaviors well complement the lack of information in *Domain B*. This is also highly related to the cold-start problem, and we give more explanation in the following. In addition to that, we will describe the impact of two proposed techniques in detail.

Impact of mitigating the cold-start problem (RQ 2). To investigate whether DS-HRNN well alleviates the cold-start problem with effectively considering multi-domain user behaviors, we conduct an additional experiment for top 20% of users who inactively behave within each domain. Table 3 shows that DS-HRNN significantly outperforms the type of existing RNN-based approaches, RNN^{SM}, by upto 29.36% in *Domain B* of the CorWeb dataset and 20.59% in the *Special* domain of the Yoochoose dataset in terms of Recall@10. Note that as previously mentioned, *Domain B* suffers from lack of information. These improvements for inactive users are much larger than those for all users, which boils down to that DS-HRNN is able to effectively predict domain-wise next user behaviors, especially for cold-start users while mitigating the cold-start problem better than others.

Impact of the domain switch-aware supplementary loss and domain switch-aware behavior regularizer (RQ 3). In order to inspect the impact of two techniques in detail, Table 4 shows the performance changes on the CorWeb dataset according to the values of loss coefficients λ_s and λ_r. Note that λ_s and λ_r can be regarded as the strength to *explicitly* recover missing direct interactions by supplementary losses and to *implicitly* recover them by behavior regularizer, respectively. Thus, when two coefficients are set to zero, DS-HRNN eventually becomes RNN^{MS}. According to the values of λ_s and λ_r, we have the two following observations: 1) Two proposed techniques are indeed beneficial to preserve local dynamics that disappeared, which enable DS-HRNN to boost the prediction accuracy. 2) Although two proposed techniques are helpful, domain change-aware supplementary loss is more beneficial than domain change-aware behavior regularizer. We attribute this difference to that the supplementary loss technique follows the explicit manner to recover missing interactions whereas the other technique does not.

Case Study – Table 5 shows our case study on two user behavior sequences in the CorWeb in order to support the necessity of two techniques. In our case study, from the urls, we observe that the last behavior is correlated with two future behaviors in both cases, especially in Case 2 where the future behavior in *Domain B* is the same as the last behavior. Note that users often revisit the web page in practice. Compared with RNN^{MS}, DS-HRNN shows better scores in terms of *rel.sim.* and higher ranks for the next true behaviors in *Domain B*. Note that *rel.sim.* enables to fairly compare the degree to which the hidden state with the last behavior is correlated with the next true behavior for the same domain. From this case study, we verify that our two proposed techniques indeed recover disconnection of local dynamics by attentively reflecting local dynamics into the recurrent and output modules of DS-HRNN.

5 CONCLUSION

User behaviors on the web become more complex since users naturally switches between various web services. To effectively model user behaviors for multiple domains, we firstly present a framework that enables a single RNN model to fully exploit sequential behavior across multiple domains. However, user behaviors across multiple domains inevitably contain practical but overlooked domain switches which hinder the performace of next user behavior prediction for each domain. To mitigate the negative impact from domain switches, we propose DS-HRNN that employs two novel domain switch-aware techniques: domain switch-aware supplementary loss and domain switch-aware behavior regularizer. In order to demonstrate the superiority of DS-HRNN, we conduct thorough experiments using two real-world datasets, and verify that DS-HRNN outperforms the state-of-the-art approaches together with the interesting case study. Moreover, DS-HRNN shows the considerable effectiveness with respect to the cold-start problem. For future work, since the proposed framework is able to take advantage of any RNNs, we also plan to leverage the content of behaviors such as text in the web pages, descriptions and images of products to more effectively model correlations among behaviors from different domains based on our proposed approach.

REFERENCES

[1] Aitor Almeida and Gorka Azkune. 2018. Predicting Human Behaviour with Recurrent Neural Networks. *Applied Sciences* 8, 2 (2018), 305.

[2] Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the GRU: Multi-task Learning for Deep Text Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 107–114.

[3] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H. Chi. 2018. Latent Cross: Making Use of Context in Recurrent Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, New York, NY, USA, 46–54.

[4] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 152–160.

[5] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 278–288. https://doi.org/10.1145/2736277.2741667

[6] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[8] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Transfer Learning for Collaborative Filtering via a Rating-matrix Generative Model. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, New York, NY, USA, 617–624. https://doi.org/10.1145/1553374.1553454

[9] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Deep Multi-Task Learning with Shared Memory for Text Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, Jian Su, Xavier Carreras, and Kevin Duh (Eds.). The Association for Computational Linguistics, 118–127. http://aclweb.org/anthology/D/D16/D16-1012.pdf

[10] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent Neural Network for Text Classification with Multi-Task Learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, Subbarao Kambhampati (Ed.). IJCAI/AAAI Press, 2873–2879. http://www.ijcai.org/Abstract/16/408

[11] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-task Learning for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, 1–10. https://doi.org/10.18653/v1/P17-1001

[12] Qiang Liu, Shu Wu, and Liang Wang. 2017. Multi-Behavioral Sequential Prediction with Recurrent Log-Bilinear Model. *IEEE Trans. on Knowl. and Data Eng.* 29, 6 (June 2017), 1254–1267.

[13] Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). 2018. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*. AAAI Press. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/schedConf/presentations

[14] Andriy Mnih and Geoffrey Hinton. 2007. Three New Graphical Models for Statistical Language Modelling. In *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*. ACM, New York, NY, USA, 641–648. https://doi.org/10.1145/1273496.1273577

[15] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *arXiv preprint arXiv:1802.08452* (2018).

[16] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 130–137.

[17] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'07)*. Curran Associates Inc., USA, 1257–1264. http://dl.acm.org/citation.cfm?id=2981562.2981720

[18] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM, New York, NY, USA, 285–295. https://doi.org/10.1145/371920.372071

[19] Elena Smirnova and Flavian Vasile. 2017. Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks. In *Proceedings of the 2Nd Workshop on Deep Learning for Recommender Systems (DLRS 2017)*. ACM, New York, NY, USA, 2–9. https://doi.org/10.1145/3125486.3125488

[20] Alessandro Suglia, Claudio Greco, Cataldo Musto, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2017. A Deep Architecture for Content-based Recommendations Exploiting Recurrent Neural Networks. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP '17)*. ACM, New York, NY, USA, 202–211.

[21] Bartlomiej Twardowski. 2016. Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 273–276. https://doi.org/10.1145/2959100.2959162

[22] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent Recommender Networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*. ACM, New York, NY, USA, 495–503. https://doi.org/10.1145/3018661.3018689

[23] Chunfeng Yang, Huan Yan, Donghan Yu, Yong Li, and Dah Ming Chiu. 2017. Multi-site User Behavior Modeling and Its Application in Video Recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, New York, NY, USA, 175–184. https://doi.org/10.1145/3077136.3080769

[24] Yu Zhang, Bin Cao, and Dit-Yan Yeung. 2010. Multi-domain Collaborative Filtering. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI'10)*. AUAI Press, Arlington, Virginia, United States, 725–732. http://dl.acm.org/citation.cfm?id=3023549.3023635

[25] Zihan Zhang, Xiaoming Jin, Lianghao Li, Guiguang Ding, and Qiang Yang. 2016. Multi-Domain Active Learning for Recommendation.. In *AAAI*. 2358–2364.

[26] Meizi Zhou, Zhuoye Ding, Jiliang Tang, and Dawei Yin. 2018. Micro Behaviors: A New Perspective in E-commerce Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, New York, NY, USA, 727–735. https://doi.org/10.1145/3159652.3159671

[27] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to Do Next: Modeling User Behaviors by time-LSTM. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. AAAI Press, 3602–3608.