Machine Learning (2019) 108:1851–1875
https://doi.org/10.1007/s10994-018-05777-9

Dynamic attention-integrated neural network for session-based news recommendation

Lemei Zhang[1] · Peng Liu[1] · Jon Atle Gulla[1]

Received: 1 April 2018 / Accepted: 25 December 2018 / Published online: 25 January 2019
© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2019

Abstract
Online news recommendation aims to continuously select a pool of candidate articles that meet the temporal dynamics of user preferences. Most of the existing methods assume that all user-item interaction history are equally importance for recommendation, which is not alway applied in real-word scenario since the user-item interactions are sometime full of stochasticity and contingency. In addition, previous work on session-based algorithms only considers user sequence behaviors within current session without incorporating users' historical interests or pointing out users' main purposes within such session. In this paper, we propose a novel neural network framework, dynamic attention-integrated neural network, to tackle the problems. Specifically, we propose a dynamic neural network to model users' dynamic interests over time in a unified framework for personalized news recommendations. News article semantic embedding, user interests modelling, session-based public behavior mining and an attention scheme that used to learn the attention score of user and item interaction within sessions are four key factors for online sequences mining and recommendation strategy. Experimental results on three real-world datasets show significant improvements over several baselines and state-of-the-art methods on session-based neural networks.

Keywords Personalized news recommendation · Recurrent neural networks · User interest modelling · Attention model

Editors: Masashi Sugiyama and Yung-Kyun Noh.

✉ Lemei Zhang
lemei.zhang@ntnu.no

Peng Liu
peng.liu@ntnu.no

Jon Atle Gulla
jon.atle.gulla@ntnu.no

[1] Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway

1 Introduction

With the rapid development of web services and e-commerce platforms, news recommender systems have become popular and are employed by many multimedia companies in recent years. They are able to cope with the information overload and to assist users in finding information matching their individual profiles learned from historical user-item interactions. However, in many real-life recommendation settings, user profiles and past activities are not available, which renders traditional recommendation methods (Adomavicius et al. 2005; Koren et al. 2009; Su and Khoshgoftaar 2009; Weimer et al. 2008) less useful. To a large extent, unprofiled users occupy a greater proportion of the total news readers, because many news websites allow users to read articles without authentication (not registered). According to the statistics performed on Cxense platform, the subscribers only take up about 20% of all users in Adresseavisen company, which is the third biggest news portal in Norway. To tackle this problem, session-based recommendation (Schafer et al. 1999) is proposed to predict the next item that the user is probably interested in based solely on implicit feedbacks, i.e., user clicks, in the current session.

To have a better understanding of user interests modelling in the session-based recommendation, we show the user clicking patterns on three topics: winter sports (vintersport), culture (kultur) and local news (nordtrondelag) over three months in the experimental dataset in Fig. 1. As can be seen from the sessions within hour of 10 and 19 in two small graphs with x-axis representing item id that the user clicked and y-axis representing user id, the clicked item sets of different users are extremely similar in their respective sessions within a time period (1 min in our paper). It means that different users across different sessions (we refer to neighbourhood sessions in the following parts) have similar interests, and they tend to focus on the most popular/emerging topics within some time period. In two small graphs, the user id and item id are consistent. Thus, we can also find that the clicked item sets of the same user across different sessions at different time slices change over time, meaning that the user interests drift at different time in a day or on different days. Besides, from the line chart in Fig. 1, we can find from the long-term click frequency of different topics that, there exist a number of periodic user interests e.g. culture topic, and continuous user interests e.g. local news and seasonal user interests e.g. winter sport, which can be valuable to recommend

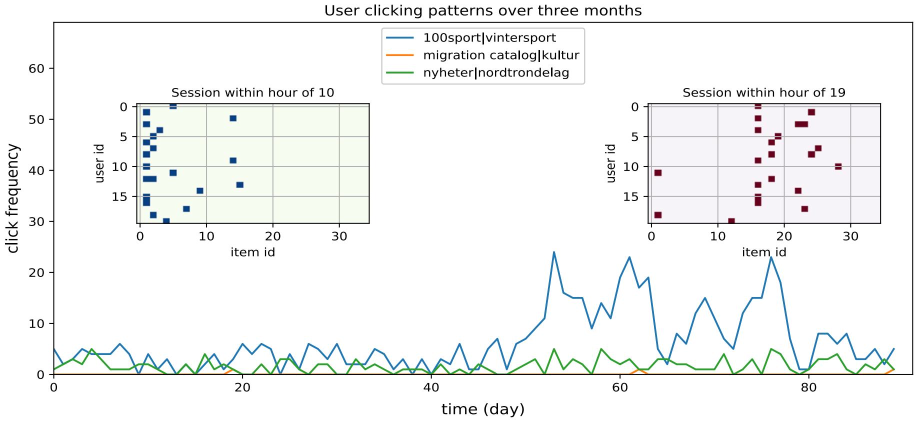


Fig. 1 User clicking patterns over three months on different topics and user clicking patterns in the neighbourhood sessions of different users within 1 min

items. Therefore, it is critical to incorporate the aforementioned factors when modelling user interest for session-based news recommendation.

Recently, Hidasi et al. (2015) apply recurrent neural networks (RNN) with Gated Recurrent Units (GRU) for session-based recommendation. The model considers the first item clicked by a user as the initial input of RNN, and generates recommendations based on it. Then the user might click one of the recommendations, which is fed into RNN next, and the successive recommendations are produced based on the whole previous clicks. Tan et al. (2016) further improve this RNN-based model by utilizing two crucial techniques, i.e., a method to account for shifts in the input data distribution and data augmentation. Despite these positive results, some problems regarding the effectiveness of the session-based recommendation method remain open: (1) they only take into account the user's sequential behavior in the current session, whereas the user's main purpose within that session is not emphasized. In other words, these methods cannot automatically select important interaction records in the user-item interaction history when recommending items. This greatly limits their application in real-world scenarios where a user accidentally clicks on wrong items or s/he is attracted by some unrelated items due to curiosity. (2) they do not incorporate the knowledge acquired on the long-term dynamics of the user interest in session-based algorithm when user profiles are available. In such case, it is reasonable to assume that the user behavior in past sessions might provide valuable information for providing recommendations in the next session.

In our paper, we propose a novel dynamic attention-integrated neural network (DAINN) to tackle the aforementioned problems for the personalized recommendation task. Specifically, DAINN models the users' dynamic interests over time by jointly incorporating users' long-term interests, user behavior sequence patterns, users' main purpose in current session, as well as public behavior mining into a unified framework. In order to improve the recommendation accuracy, dynamic topic modelling (Blei and Lafferty 2006) and convolutional neural network (CNN) sentence model (Kim 2014) are adopted to effectively learn the item semantic embedding. More importantly, to handle diverse variance of users' clicking behavior, we introduce a novel attention scheme that would dynamically assign influence factors on recent models based on the users' spatio-temporal reading characteristics. We applied our model to several real data sets and the experimental results demonstrate promising and reasonable performance of our approach.

This paper makes the following contributions:

- We propose a dynamic attention-integrated neural network (DAINN) to model users' dynamic interests over time in a unified framework for personalized session-based news recommendation.
- The proposed model can jointly exploit users' long-term interests, user behavior sequence patterns, users' main purpose in current session, as well as public behavior mining to model users' preference. In addition, item semantic embedding learned from CNN sentence model is adopted to further improve the recommendation accuracy.
- To handle the diverse variance of users' clicking behavior, a novel attention scheme is proposed, which considers the spatio-temporal reading characteristics of users.
- We apply DAINN to three real-world datasets with extensive experiments. The results show that DAINN achieves substantive gains over state-of-the-art deep learning based methods for recommendation. Specifically, DAINN outperforms baselines by 3 to 5% on F1 score and 2 to 5% on MRR.

The remainder of the paper is organized as follows. Section 2 introduces the related work on news recommendation, deep recommender system and attention model. In Sect. 3, we formally define our problem and present DAINN model. We describe the data sets, experiment settings and the prior information we use in Sect. 4. Section 5 shows a comprehensive experiment evaluation. Finally, we present the conclusions and future work in Sect. 6.

2 Related work

2.1 News recommendation

2.1.1 Traditional methods

News recommendation aims to recommend to users the news that match their personal interests best (Li et al. 2010). As a popular service and an important way to retain users, industry puts much efforts in news recommendation researches (Das et al. 2007). Several adaptive news recommending systems, such as Google News and Yahoo! News provide personalized news recommendation services for a substantial amount of online users. Existing news recommender systems can be roughly categorized into three groups: collaborative filtering, content-based filtering and hybrid methods. The first one makes use of news ratings by users to provide recommendation services, and they are content-free. In practice, most collaborative filtering systems are constructed based on users' past rating behaviors, either using a group of users similar to the given user to predict news ratings (Sarwar et al. 2001), or modelling users' behaviors in a probabilistic way (Hofmann 2004). However, collaborative filtering is ineffective for cold-start problem. Content-based methods try to sequentially find newly-published articles similar to the user's reading history in terms of content. Generally speaking, news content is often represented using vector space model (e.g., TF-IDF) (Jurafsky and Martin 2014), or topic distributions obtained by language models (e.g., PLSI and LDA), and specific similarity measurements are adopted to evaluate the relatedness between news articles. However, in some scenario, simply representing the user's profile information by a bag of words is insufficient to capture the exact reading interest of the user. Recently, hybrid solutions are attracted more attentions to improve recommendation results. Representative examples include Rao et al. (2013), in which the inability of collaborative filtering to recommend news items is alleviated by combining it with content-based filtering.

2.1.2 Sequential-based methods

Sequential recommender is based on Markov chains which utilize sequential data by predicting users' next action given the last action (Shani and Brafman 2005; Zimdars et al. 2001). Zimdars et al. (2001) propose a sequential recommender based on Markov chains and investigate how to extract sequential patterns to learn the next state using probabilistic decision-tree models. Shani and Brafman (2005) present a Markov Decision Processes (MDP) aiming to provide recommendations in a session-based manner and the simplest MDP boil down to first-order Markov chains where the next recommendation can be simply computed through the transition probabilities between items. Bamshad et al. (2002) study different sequential patterns for recommendation and find that contiguous sequential patterns are more suitable for sequential prediction task than general sequential patterns. Yap et al. (2012) introduce a new Competence Score measure in personalized sequential pattern mining for next-item recommendations. Chen et al. (2012) model playlists as Markov chains, and propose logistic Markov Embeddings to learn the representations of songs for playlists prediction. A major issue with applying Markov chains in the session-based recommendation task is that the state space quickly becomes unmanageable when trying to include all possible sequences of potential user selections over all items.

Recently, several studies have been done to use neural network based models including deep learning techniques for recommendation tasks. Yu et al. (2016) represent a basket acquired by pooling operation as the input layer of RNN, which outperforms the state-of-the-art methods for next basket recommendation. Song (2016) propose a multi-rate Long Short-Term Memory (LSTM) with considering both long-term static and short-term temporal user preferences for commercial news recommendation. Hidasi et al. (2015) propose to use RNN to model whole sequences of session click IDs. In a later work, they (Hidasi et al. 2016) extend their previous work by combining rich features of clicked items such as item IDs, textual descriptions, and images. They use different RNNs to represent different types of features and train those networks in a parallel fashion. More recently, with the ability to express, store and manipulate the records explicitly, dynamically and effectively, external memory networks (EMN) (Sukhbaatar et al. 2015) have shown their promising performance for many sequential prediction tasks, such as question answering (QA) (Kumar et al. 2016), natural language transduction (Grefenstette et al. 2015), and recommender system (Chen et al. 2018). Chen et al. (2018) proposed a novel framework integrating recommender system with external User Memory Networks which could store and update users' historical records explicitly. Huang et al. (2018) proposed to extend the RNN-based sequential recommender by incorporating the knowledge-enhanced Key-Value Memory Network (KV-MN) for enhancing the representation of user preference. Our work is relevant to Hidasi et al. (2016) in that we combine features of different type for better session-based recommendation. However, our method uses a totally different model (DAINN) and encoding method, which provide improved accuracy while simplify feature engineering steps.

2.2 Deep recommender system

Deep learning has been successfully employed in computer vision (Krizhevsky et al. 2012), speech recognition (Graves et al. 2013), and several other application domains (LeCun 2015). Among these applications, convolutional neural networks (CNN) and recurrent neural networks (RNN) are two most popular deep learning models. Other deep learning models include auto-encoders, Restricted Boltzman Machines (RBMs), and fully connected networks with multiple hidden layers (LeCun 2015). In recent years, deep learning methods have also been shown to be promising in the area of recommender systems. One of the first related methods along this direction was presented by Salakhutdinov et al. (2007), in which several layers of RBMs are stacked together to deliver a better accuracy than a CF algorithm using singular value decomposition. Deep Models have been used to extract features from unstructured content such as music or images that are then used together with more conventional collaborative filtering models. Wang et al. (2015) introduced a more generic approach whereby a deep network is used to extract generic content-features from any types of items, these features are then incorporated in a standard collaborative filtering model to enhance the recommendation performance. Van den Oord et al. (2013) proposed a somewhat similar hybrid method exploiting a convolutional deep network to learn features from content descriptions of songs, which are then used in a CF model to tackle the data sparsity problem. The difference is that they use CNNs for feature learning rather than auto-encoders. Our method also uses CNNs and content features, but our model allows capturing temporal patterns, which is important for sequential nature of session clicks.

Though a growing number of publications on session-based recommendation focus on RNN-based methods, unlike existing studies, we propose a novel dynamic neural attentive recommendation model that combines the user's sequential behavior and main purpose in the current session as well as the users' historical interests, which to the best of our knowledge, is not considered by existing researches.

2.3 Attention model

Attention is a mechanism to flexibly selecting the reference part of context information, which can facilitate global learning (Bahdanau et al. 2014; Xu et al. 2015). Attention model was originally proposed in machine translation tasks to deal with the issue for encoder-decoder approaches that all the necessary information should be compressed into the fix-length encoding vector (Bahdanau et al. 2014). Soon after the use on language, attention model is leveraged on image caption task (Xu et al. 2015) where the salient part of an image is automatically detect and based on that the model could generate high-quality description of the image. Then, the attention model is leveraged in various tasks. Yang et al. (2016) utilized attention to capture hierarchical patterns of documents from word to sentence and finally to the whole document. Yang et al. (2016) took attention on question text and extracted the semantically related parts between question-answer pairs. Other attention-based work includes natural language parsing (Vinyals et al. 2015) and text classification (Zhang et al. 2016).

Recently, the attention model has also been used in recommender systems and achieves better performance in many recommendation scenarios. To model the different impacts of a user's diverse historical interests on current candidate news, Wang et al. (2018) designed an attention module to dynamically calculate a user's aggregated historical representation. Pei et al. (2017) extended recurrent networks for modelling user and item dynamics with a novel gating mechanism, which adopts the attention model to measure the relevance of individual time steps of user and item history for recommendation. Li et al. (2017) explored a hybrid encoder with an attention model to capture both the user's sequential behavior and main purpose in the current session. Specifically, they involved an item-level attention mechanism which allows the decoder to dynamically select and linearly combine different parts of the input sequence. Our work is highly built upon the work Li et al. (2017). The novelty lies in the idea of incorporating users' spatio-temporal reading characteristics into the dynamic attention model for the session-based news recommendation. As far as we know, this is the first attempt to capture the diverse variance of users' clicking behavior with a dynamic hybrid attention scheme.

3 Methodology

In this section, we propose a novel dynamic attention-integrated neural network (DAINN) for session-based news recommendation. Firstly, the problem is defined, including the relevant general terms and notations. Then we give the details about the unified recommendation framework, which includes user long-term interest modelling, temporal context mining, session-based public behavior mining, dynamic attention learning. As shown in Fig. 2, the proposed DAINN model can be regarded as an interest network by considering the four factors jointly for learning users' dynamic preferences.

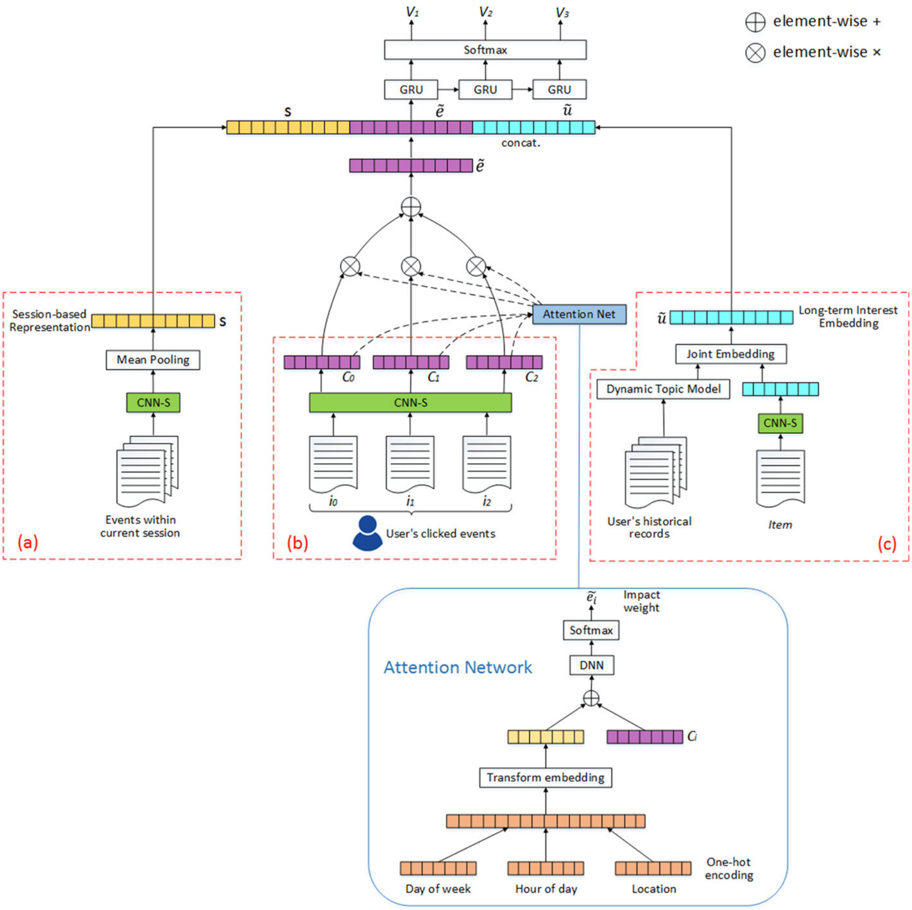


Fig. 2 The unified framework for the personalized news recommendation via dynamic attention-integrated neural network

3.1 Problem definition

3.1.1 Notation

Throughout this paper, all vectors are column vectors and are denoted by bold lower case letters (e.g., $\boldsymbol{x}$ and $\boldsymbol{y}$), while matrices are represented by bold upper case letters (e.g., $\boldsymbol{X}$ and $\boldsymbol{M}$). The ith row of a matrix $\boldsymbol{X}$ is given by $\boldsymbol{X}_{i\cdot}$, while $\boldsymbol{X}_{\cdot j}$ represents the jth column. We use calligraphic letters to represent sets (e.g., $\mathcal{V}$ and $\mathcal{E}$). Table 1 summarizes the notations of frequently used variables.

3.1.2 Session-based recommendation

Session-based recommendation is the task of predicting what a user would like to click next when his/her current sequential transaction data is given. Here we give a formulation of the session-based recommendation problem.

Table 1 Notations used in the paper

Symbol	Description
$\boldsymbol{x}_i$	Representation of one clicked item in a session
n_s	The number of events in one session
m	The number of candidate items
n	The number of words in the input sentence
$\mathcal{V}$	The set of word vocabulary
N_w	The total number of words in vocabulary $\mathcal{V}$
d	Dimension of word embedding
$\boldsymbol{c}$	Representation of semantic embedding
w	Sliding window
n_u	The total number of clicked events within one session for user u
$\boldsymbol{s}$	Session-based representation
θ	Parameters of DAINN framework
$\boldsymbol{T}_p$	Parameter matrix of the softmax layer in GRU
N_u	The number of user's historical interested topics
λ	Time decay parameter
$\boldsymbol{T}_c, \boldsymbol{T}_u$	Transformation matrix in user long-term modelling
D_e, D_c	Dimensionality of the learned user topics representation and the textual embedding
N_c	Normalization parameter in user long-term modelling
D_d, D_h, D_l	The number of days in a week, the number of hours in a day and the number of locations in dataset
$\tilde{\boldsymbol{e}}_t$	The embedding after attention network of user u at timestamp t
$\tilde{\boldsymbol{u}}$	The embedding after user long-term modelling of user u
$\boldsymbol{v}$	The word distribution

Let $[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_s-1}, \boldsymbol{x}_{n_s}]$ be a click session, where $\boldsymbol{x}_i \in \mathcal{I}\,(1 \le i \le n_s)$ is the representation of one clicked item out of a total number of m candidate items. We build a model $\boldsymbol{\mathcal{F}}$ so that for any given prefix of the click sequence in the session, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t]$, $1 \le t \le n_s$, we get the output $\boldsymbol{y} = \boldsymbol{\mathcal{F}}(\boldsymbol{X})$, where $\boldsymbol{y} = [y_1, y_2, \ldots, y_{m-1}, y_m]$. We view $\boldsymbol{y}$ as a ranking list over all the next items that can occur in that session, where y_j $(1 \le j \le m)$ corresponds to the recommendation score of item j. Since a recommender typically needs to make more than one recommendations for the user, thus the top-k $(1 \le k \le m)$ items in $\boldsymbol{y}$ are recommended.

3.2 Dynamic attention-integrated neural network

3.2.1 Overview

To improve the recommendation performance in news domain and address session- based recommendation problems, we proposed a novel dynamic attention-integrated neural network (DAINN). The basic idea of our model is to build a unified represent- ation of the current user, and then generate predictions on the user's next possible event with it. The representation should take into account various potential factors that influence user's next decision. As shown in Fig. 2, the input of GRU is a joint output from three components, namely session-based public behavior mining, dynamic attention learning and user long-term interest modelling represented as (a), (b) and (c) respectively. The basic input is the sequence $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n]$ where $\boldsymbol{x}_i$ denotes the word-level representation of item i. Component (a) transfers the input sequence $\boldsymbol{X}_p = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_p}]$ collected from users within a predefined sliding window ω except current user u, into the representation $\boldsymbol{s}$ of public behavior pattern. Component (c) learns from user u's historical records and outputs the representation $\tilde{\boldsymbol{e}}$ of user's long-term interest pattern. Meanwhile component (b) converts the input sequence $\boldsymbol{X}_s = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_s}]$ of user u's current session into the high dimensional representation $\tilde{\boldsymbol{u}}$ with user's current purpose, along with the attention weight at time t (represented as s_t). Finally, the concatenation of the three representations is fed into GRU to generate top-k items with the highest possibilities that user u will click next. One should be clarified that CNN for semantic embedding models, denoted as CNN-S in Fig. 2, share parameters in three components. CNN-S is adopted to extract semantic information from simple word-level representations of inputs, and its output is denoted as $\boldsymbol{c}$ in our paper.

In the following part of this section, we first describe the CNN-S model for semantic embedding in Sect. 3.2.2 used in each component. Then we introduce the session-based public behavior mining which is used to extract neibourhood session patterns from public users of component (a) in Sect. 3.2.3, user's long-term interest modelling of component (c) in Sect. 3.2.4 and dynamic attention learning which is used to extract user's main purpose within current session of component (b) in Sect. 3.2.5. The learning objective is introduced in Sect. 3.2.6 and finally in Sect. 3.2.7, the top-k items generation process is explained.

3.2.2 CNN for semantic embedding

To model the textual content of the document, traditional methods including bag-of-words features (Agarwal et al. 2009; Melville et al. 2002), e.g. TF-IDF feature or Naive Bayes and unsupervised learning objective (Gopalan et al. 2014; Wang et al. 2011), e.g. topic models, are based on counting statistics which ignore word orders and suffer from sparsity and poor generalization performance. A more effective way to model the text is to represent each sentence in a given corpus as a distributed low-dimensional vector. Recently, inspired by the success of applying convolutional neural networks (CNN) in the field of computer vision (Krizhevsky et al. 2012), researchers have proposed many CNN-based models for semantic embedding (Kim 2014; Zhang et al. 2015).[1] In this subsection, we introduce a typical type of CNN architecture, namely Kim CNN (Kim 2014).

Figure 3 illustrates the architecture of Kim CNN. In Fig. 2, Kim CNN are denoted as CNN-S. Let $W_{1:n}$ be the raw input of a sentence of length n, and $\boldsymbol{x} = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{1\times n}$ be the word-level representation vector of the input sentence, where $x_i \in \mathbb{R}$ is the index of the ith word in vocabulary $\mathcal{V}$ in the sentence. We can get the word embedding of the ith word through word2vec pre-trained model $\boldsymbol{w}_i = \boldsymbol{\mathcal{H}}(x_i; \mathcal{V})$, $\boldsymbol{w}_i \in \mathbb{R}^{d\times 1}$, where d is the dimension of word embeddings. Thus, we can get $\boldsymbol{W}_{1:n} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n] \in \mathbb{R}^{d\times n}$, the word embedding matrix of the input sentence. A convolution operation with filter $\boldsymbol{h} \in \mathbb{R}^{d\times l}$ is then applied to the word embedding matrix $\boldsymbol{W}_{1:n}$, where l $(l \leq n)$ is the window size of the filter. Specifically, a feature $\boldsymbol{d}_i$ is generated from a sub-matrix $\boldsymbol{W}_{i:i+l-1}$ by

[1] Researchers have also proposed other types of neural network models for semantic embedding such as recurrent neural networks (Tai et al. 2015), recursive neural networks (Socher et al. 2013), and hybrid models (Lai et al. 2015). However, CNN-based models are empirically proven to be superior than others (Hong and Fang 2015), since they can detect and extract specific local patterns from sentences due to the convolution operation. To keep our presentation focused, we only discuss CNN-based models in this paper.

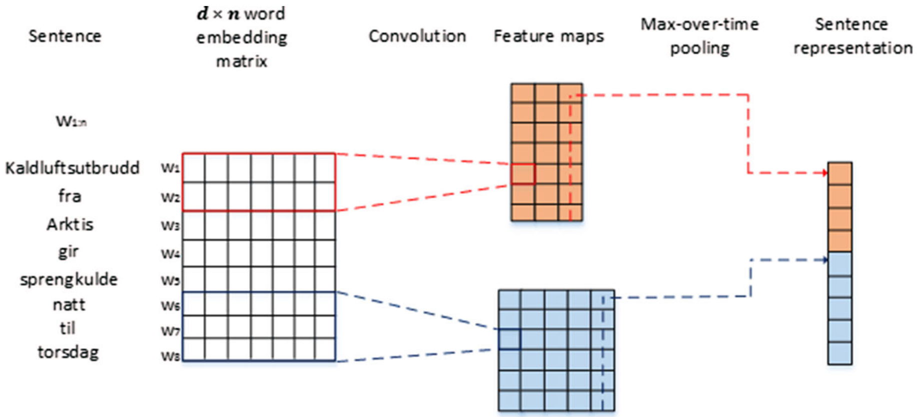


Fig. 3 A typical architecture of CNN for semantic embedding (Kim 2014)

$$\boldsymbol{d}_i = f(\boldsymbol{h} * \boldsymbol{W}_{i:i+l-1} + b) \tag{1}$$

where f is a non-linear transformation function such as the hyperbolic tangent (tanh) $f(z) = (exp(z) - exp(-z))/(exp(z) + exp(-z))$, $*$ is the convolution operator, and $b \in \mathbb{R}$ is a bias. After applying the filter to every possible position in the word embedding matrix, a feature map

$$\boldsymbol{D} = \left[\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_{n-l+1}\right] \tag{2}$$

is obtained, then a max-over-time pooling operation is used on feature map $\boldsymbol{D}$ to identify the most significant feature:

$$\boldsymbol{c} = max\{\boldsymbol{D}\} = max\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_{n-l+1}\} \tag{3}$$

One can use multiple filters (with varying window sizes) to obtain multiple features, and these features are concatenated together to form the representation of the textual content.

3.2.3 Session-based public behavior mining

As described in Fig. 1, user clicking patterns across neighbourhood sessions with different users within some time periods are extremely similar. Besides, many recent works (Hidasi et al. 2015; Wu and Yan 2017) have proved the efficiency of adopting session-based methods on especially non-profile users. According to the statistics on our experimental dataset, users with historical records (also known as subscribers) take up less than 20% of total number of users. Thus, inter- and intra-session information is essential for recommendations.

Assuming that $\boldsymbol{X}_p = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_p}]$ is a sequence of events that clicked within a predefined sliding window ω which is the time period before the current time t. n_p denotes the number of items within window ω. Each $\boldsymbol{x}_i$ represents the word-level item representation of the user excluding the current user u. As illustrated in Fig. 2a, $\boldsymbol{X}_p$ is firstly put into CNN-S model to obtain the textual semantic embedding of these items according to Eq. (3) denoted as $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{n_p}]$. Then we use mean-pooling through horizontal axis as user u's session representation$\boldsymbol{s}$

$$\boldsymbol{s} = \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{c}_j \tag{4}$$

If we do not consider the attention network and user u is a newly arrived user which has no historical record, we only consider item $\boldsymbol{x}_t$ that user u clicks at current time t. Then the semantic representation denoted as $\boldsymbol{c}_1$, of $\boldsymbol{x}_t$ can be acquired through CNN-S model. After that, the concatenation of embedded session-based representation $\boldsymbol{s}$ and semantic representation $\boldsymbol{c}_1$, $\boldsymbol{s} \oplus \boldsymbol{c}_1$, is sent through one or multiple layers of the Gated Recurrent Unit (GRU) (Cho et al. 2014) which is the simplified version of Long Short-Term Memory (LSTM) networks but still maintains all their properties. In GRU unit, the activation h_t at time t is a linear interpolation between the previous activation h_{t-1} and the candidate activation $\tilde{h}_t$:

$$\boldsymbol{h}_t = (1 - z_t)\boldsymbol{h}_{t-1} + z_t\tilde{\boldsymbol{h}}_t \tag{5}$$

where an update gate z_t decides how much the unit updates its activation, or content. The update gate is computed by

$$z_t = \sigma(\boldsymbol{W}_z\tilde{\boldsymbol{x}}_t + \boldsymbol{U}_z\boldsymbol{h}_{t-1}) \tag{6}$$

This procedure of taking a linear sum between the existing state and the newly computed state is similar to the LSTM unit. The GRU, however, does not have any mechanism to control the degree to which its state is exposed, but exposes the whole state each time. The candidate activation $\tilde{h}_t$ is computed similarly to that of the traditional recurrent unit but slightly different from Cho et al. (2014)

$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W}\tilde{\boldsymbol{x}}_t + \boldsymbol{U}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})) \tag{7}$$

where $\boldsymbol{r}_t$ is a set of reset gate and $\odot$ is an element-wise multiplication, and $\tilde{\boldsymbol{x}}$ is the output from previous layer or $\boldsymbol{s} \oplus \boldsymbol{c}$. We experimented on both formulations to compute $\tilde{\boldsymbol{h}}_t$ and they performed as well as each other. When $\boldsymbol{r}_t$ is close to 0, the reset gate effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state. The reset gate can be computed as

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}\tilde{\boldsymbol{x}}_t + \boldsymbol{U}_r\boldsymbol{h}_{t-1}) \tag{8}$$

The output of GRU at timestamp t can be denoted as $\boldsymbol{o}_t = \boldsymbol{h}_t$. Inspired by the work of Pan et al. (2016), we formulate our recommendation problem as a coherence loss, where the log probability of the recommendation is given by the sum of log probabilities over the clicked items as shown below

$$\mathcal{L}_{rec}(\boldsymbol{X}, \boldsymbol{v}) = -\log P(\boldsymbol{v}|\boldsymbol{X}) = \sum_{t=1}^{n_s} -\log P(\boldsymbol{v}_t|\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{t-1}; \theta) \tag{9}$$

where $\{\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{n_s}\}$ is the sequentially predicted items. Here, $\tilde{\boldsymbol{x}}_i$ is corresponding to the representation of item i. $\boldsymbol{v}_t$ is the words distribution of next possible item. θ are the parameters of our framework, including parameters of CNN-S model and GRU model. By minimizing the above loss, the user's interests within and across sessions can be described dynamically. Here, a softmax layer is applied after GRU layer to produce a probability distribution over all the N_w words in the vocabulary as

$$P(\boldsymbol{v}_t|\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{t-1}; \theta) = \frac{\exp\{\boldsymbol{T}_p\boldsymbol{h}_t\}}{\sum_{j=1}^{N_w}\exp\{\boldsymbol{T}_p\boldsymbol{h}_t\}} \tag{10}$$

where $\boldsymbol{T}_p$ is the parameter matrix of the softmax layer in GRU.

3.2.4 User long-term interest modelling

Although texts have semantic information, they cannot reflect users' broad interest directly (Cui et al. 2014). To represent texts and users' interest in a common space, as shown in Fig. 2c, we jointly learn the relevance between text semantic embedding and user interested topics. Specifically, we first conduct some online topic modelling approach on all the users' historical behavior streams (e.g., news clicking streams or song listening streams) to build a shared user topic space and learn the topical distribution for each user. Then we aggregate the topic distributions of each user's real-time behavior streams to derive the representation of user interested topics at the current time, where a time decay (Ding and Li 2005) is used to weight the behavior streams. Therefore, the user's interested topics can be defined as in Eq. (2).

$$\boldsymbol{u} = \frac{1}{N_u} \sum_{i \in \mathcal{B}_u} \boldsymbol{m}_i \cdot e^{-\lambda |t - t_i|} \tag{11}$$

where $\boldsymbol{m}_i$ denotes representation of a user's interested topics of the ith behavior, $\mathcal{B}_u$ is the user's historical behaviors, $|t - t_i|$ indicates the time difference between the current time and the post time of user behavior i. N_u is the number of user's historical interested topics and λ is the time decay parameter. In this paper, topics are extracted by Dynamic Topic Model introduced in Greene and Cross (2016) and $\boldsymbol{m}_i$ is the word-level representation of topic i.

To project the textual semantic embedding and user interested topics into a common space, we adopt two transformation matrices, $\boldsymbol{T_c} \in \mathbb{R}^{D_e \times D_c}$ and $\boldsymbol{T_u} \in \mathbb{R}^{D_e \times D_u}$, where D_u and D_c is the dimensionality of the learned user topics representation and textual embedding respectively. To measure the relevance between textual semantic embedding and the user interested topics, one direct way is to calculate the distance between them. We integrate the textual semantic embedding of a user's clicking list in Eq. (3), and the distance loss is defined in Eq. (4):

$$\tilde{\boldsymbol{u}} = \frac{1}{N_c} \sum_{c \in \mathcal{T}_u} \boldsymbol{c} \cdot e^{-\lambda |t - t_c|} \tag{12}$$

$$\mathcal{L}_{long}(\boldsymbol{\mathcal{U}}, \tilde{\boldsymbol{\mathcal{U}}}) = \sum_{\boldsymbol{u} \in \boldsymbol{\mathcal{U}}, \tilde{\boldsymbol{u}} \in \tilde{\boldsymbol{\mathcal{U}}}} ||\boldsymbol{T_u} \cdot \boldsymbol{u} - \boldsymbol{T_c} \cdot \tilde{\boldsymbol{u}}||_F^2 \tag{13}$$

where $\mathcal{T}_u$ is the textual semantic embedding vectors of the clicked news for user u. $|t - t_c|$ indicates the time difference between the current time and the post time when the user clicks the specific news/song. N_c is a normalization parameter. One need to be noticed that, if user long-term interests can be achieved, for personalized recommendation tasks, CNN-S model in Fig. 2a–c parts will share parameters for user u.

3.2.5 Dynamic attention learning

Given user u with clicked items $\{i_1, i_2, \ldots, i_{n_s}\}$ within session s, and his/her learned contextual representation after CNN-S can be defined as $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{n_s}\}$. To represent user u's attention at timestamp t, one can simply average all the embeddings of his/her clicked items:

$$\tilde{\boldsymbol{c}}_t = \frac{1}{n_s} \sum_{k=1}^{n_s} \boldsymbol{c}_k \tag{14}$$

However, user's interests are full of stochasticity and contingency, and user's clicked items supposed to have different impacts on the next possible clicking item. Specifically, our attention measurement scheme is mainly constructed based on threefold factors:

- *Day of Week* users read different topics of news at different week days, for example, during a working day or at the weekend, while relaxing.
- *Hour of day* As illustrated in Fig. 1, user's interested topics may vary over time across day. For instance, a user may tend to read more financial news in the morning than in the afternoon, while s/he reads more sports or entertainment news at night.
- *Location* According to the analysis of address a dataset, we find that users incline to read news happening around them. For example, a user from Oslo reads more news occurred in Oslo than news occurred in other regions.

In order to incorporate these three aspects, we first use the one-hot representation to denote the three factors. Specifically, we take binary vectors $\boldsymbol{r}_d \in R^{D_d}$, $\boldsymbol{r}_h \in R^{D_h}$ and $\boldsymbol{r}_l \in R^{D_l}$, where only the value of the column corresponding to the presented day, hour and location are set as 1 and the values for other columns are 0. D_d, D_h and D_l represent the number of days in a week, the number of hours in a day and the number of locations in dataset respectively. Then, the three vectors are concatenated as $\boldsymbol{r}_t = [\boldsymbol{r}_d; \boldsymbol{r}_h; \boldsymbol{r}_l]$. To learn the three factors and item's representation $\boldsymbol{c}_i$ together, one ordinary way is the simple concatenation strategy as $\boldsymbol{e}_t = [\boldsymbol{c}_t; \boldsymbol{r}_t]$. However, we argue that factor embedding and item embedding are learned by different methods, which means they are in different representation space. Thus, we introduce the transformed embeddings

$$\tilde{\boldsymbol{r}}_t = g(\boldsymbol{r}_t) \tag{15}$$

where $g(\cdot)$ is the transformation function, and can be either linear

$$g(\boldsymbol{r}_t) = \boldsymbol{T}\boldsymbol{r}_t \tag{16}$$

or non-linear

$$g(\boldsymbol{r}_t) = sigmoid(\boldsymbol{T}\boldsymbol{r}_t + \boldsymbol{b}) \tag{17}$$

where $\boldsymbol{T} \in R^{D_{\tilde{r}} \times D_r}$ is the trainable transformation matrix and $\boldsymbol{b} \in R^{D_{\tilde{r}} \times 1}$ is the trainable bias. Since the transformation is continuous, it can map factor embeddings to item space while preserving their original spacial relationship. We therefore can concatenate these two embeddings as $\boldsymbol{e}_t = [\boldsymbol{c}_t; \tilde{\boldsymbol{r}}_t]$ at timestamp t.

Inspired by the work in Wang et al. (2018), we use an attention network to model the different impacts of user's clicked news $\boldsymbol{c}_t$. The attention network is illustrated in the bottom part of Fig. 3. Different from Wang et al. (2018), we not only consider the clicking patterns within current session, but also integrate various influential factors into the attention model. Specifically, for user u's clicked news representation $\boldsymbol{c}_t$ at timestamp t and factor representation $\tilde{\boldsymbol{r}}_t$, after concatenation of their embeddings, we apply a DNN $\mathcal{G}$ as the attention network and the softmax function to calculate the normalized impact weight:

$$s_t^i = softmax\left(\mathcal{G}\left(\boldsymbol{e}_t^i\right)\right) = \frac{\exp\left(\mathcal{G}\left(\boldsymbol{e}_t^i\right)\right)}{\sum_{k=1}^{N_u} \exp\left(\mathcal{G}\left(\boldsymbol{e}_t^i\right)\right)} \tag{18}$$

The attention network $\mathcal{G}$ receives concatenation embeddings as input and outputs the impact weight. Then the embedding of user u at timestamp t can be calculated as the weighted sum of his clicked news embeddings:

$$\tilde{\boldsymbol{e}}_t = \sum_{k=1}^{N_u} s_t^i \boldsymbol{e}_t^i \tag{19}$$

We will demonstrate the efficacy the attention network in the experiment section.

3.2.6 Unified recommendation framework

Recall that in Sect. 3.2.1, we formulate our recommendation problem as a coherence loss in Eq. (13) with respect to the input item representation and the output words distribution. If we also consider user's historical records as described in Sect. 3.2.3, given a user's current interested topics u, we can formulate our recommendation problem as

$$\mathcal{L}_{rec}(\boldsymbol{u}, \boldsymbol{v}) = -\log P(\boldsymbol{v}|\boldsymbol{u}, X) = \sum_{t=1}^{n_s} -\log P(\boldsymbol{v}_t|\boldsymbol{u}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}; \theta) \tag{20}$$

where θ represents not only the parameters of GRU and session-based CNN sentence network, but also $\boldsymbol{T_u}$, $\boldsymbol{T_c}$ of user long-term interest model. The input of GRU layer is the concatenation of session-based representation $\boldsymbol{s}$, output of attention model $\tilde{\boldsymbol{e}}$ and the user long-term interest embedding $\tilde{\boldsymbol{u}}$, denoted as $\boldsymbol{s} \oplus \tilde{\boldsymbol{e}} \oplus \tilde{\boldsymbol{u}}$. By minimizing the above loss, the user interest evolvement can be described dynamically, which makes the recommendation more coherent and reasonable. Finally, we can obtain the objective function below

$$\mathcal{L} = \sum_{\boldsymbol{u} \in \mathcal{U}} \mathcal{L}_{rec}(\boldsymbol{u}, \boldsymbol{v}) + \lambda_1 \mathcal{L}_{long}(\boldsymbol{\mathcal{U}}, \tilde{\boldsymbol{\mathcal{U}}}) + \lambda_2 ||\theta||_2^2 \tag{21}$$

where λ_1 is the trade-off parameter for these objectives, and λ_2 is the coefficient of the weight decay term. By optimizing the above overall loss function in a unified framework, our proposed method achieves dynamic news recommendation with considering inter- and intra-session modelling, user interest modelling, as well as dynamic attention learning.

3.2.7 Recommending top-K items

Given a target user u with the request time t, in order to recommend top-K items that user u would like to choose, we compute the ranking score with respect to the predicting words distribution and item word-level distribution as in Eq. (22)

$$S(\boldsymbol{v}_i, \boldsymbol{v}_j, t) = \boldsymbol{v}_i \cdot \boldsymbol{v}_j = \sum_{k=1}^{N_w} v_{ik} \cdot v_{jk} \tag{22}$$

where $\boldsymbol{v}_i = [v_{i1}, \ldots, v_{iN_w}]$, N_w is the number of vocabulary. Since the effectiveness of news articles are very short (usually less than 7 days), the candidate items are limited for target users when performing Eq. (22). In other words, we can filter candidate items according to their publication time before calculate items' ranking score, and thus avoiding computing all possible items in database.

Table 2 Some statistics of the datasets

Dataset	Adressa	Last.fm	Weibo-Net-Tweet
#sessions	9,211,140	73,273	2,126,697
#users	4,805,071	2501	1,776,950
#events	113,579,695	580,393	23,755,810
#items	48,486	7899	300,000
#events train	104,368,555	507,120	21,269,113
#events test	9,211,140	73,273	2,126,697
#events per session	12	8	11

4 Experimental setup

4.1 Datasets

We used three datasets from different areas for our experiments, namely Adressa, Last.fm and Weibo-Net-Tweet. The first is the Adressa 16G dataset[2] which contains 93,948 news articles, 398,545 readers, and about 113 million events over a 90-days period (Kim 2014). Each of these events represent that a user read a particular news article. As preprocessing, we filtered sessions with less than 3 events for a user. Besides, we removed the records that users visited the news front page for there are no articles related information within the events. In order to evaluate our model's generality, we adopted Last.fm provided by Schedl (2016), which contains 10 weeks of log data between 1/1/2013 and 11/3/2013.[3] To enrich the content information for the dataset, we also used Last.fm API[4] to collect artist information to improve the recommendation accuracy. The third dataset is provided by Jing et al. (2013) from Sina Weibo.com,[5] which includes in total 1.7 million users and 300 thousand microblogs. We perform the similar preprocessing procedure on the other two types of datasets, including getting rid of the session with less than 3 events and removing the duplicate records. The characteristics of the datasets are summarized in Table 2.

To split users' historical logs into sessions, for Adressa dataset, it contains tags to represent the start and end of a session. As for Last.fm and Weibo-Net-Tweet datasets, following Zheleva et al. (2010) and Baur et al. (2012), we use the time gap approach to generate sessions. If the gap between two post items is less than 30 min for user u, they belong to the same session. Otherwise, they will be separated into two sessions.

The testing set is build with the last event of each session of each user. The remaining events form the training set. Besides, we also leave the last event of each session of each user from training set as a validation set, which is used for hyper-parameter selection during each iteration in training procedure. In order to test the user's long-term interests influence on our recommendation approach, we also selected users with historical records, namely user profile, from these three datasets to do the evaluation. The rest users without user profile are experimented as cold start problem in the following section.

[2] http://reclab.idi.ntnu.no/dataset/.

[3] http://www.cp.jku.at/datasets/LFM-1b/.

[4] https://www.last.fm/api/show/artist.getInfo.

[5] https://www.aminer.cn/influencelocality.

4.2 Evaluation metrics

Based on temporally ordered lists of read/played items, our objective is to correctly predict the next item a target user will likely read/play. The ground truth at a particular time step is therefore represented by a single user-item tuple. To present the user with adequate recommendations, the target item should be among the top few recommended items. Since we are interested in measuring top-k recommendation instead of rating prediction, we measure the performance by looking at the Recall@k, Precision@k, F1 score and MRR@k, which are widely used for evaluating top-k recommender systems.

- MRR@k (Mean Reciprocal Rank) is defined as the average of the reciprocal ranks of the desired items (Voorhees 1999). The rank is set to zero if it is above k.
- Precision@k is defined as the proportion of recommended items in the top-k set that are actually consumed in the next event.
- Recall@k is defined as the proportion of the items actually consumed in the next event among the top k items recommended.
- F1 score is the harmonic mean between recall and precision values and can be denoted as $F_1 = 2 * Precision@k * Recall@k/(Precision@k + Recall@k)$ (Powers 2011).

In recommendation performance experiment, we vary k to 5, 10, 20 to test top-k recommendation efficiency. In other experiments, we set k = 20, as it appears desirable from a user's perspective to expect the target among the first 20 items (Hidasi et al. 2015).

4.3 Baselines

To validate the effectiveness of DAINN, we compared our model with the following session-based recommendation methods.

- Popular-based Method (POP): This method recommends items with the largest number of interactions by the users.
- Item KNN: Item KNN is a simple, yet effective method, which is widely deployed in practice. In this method, two item are considered similar if they co-occur frequently in different sessions. In our situation, we recommend items based on cosine similarity between different sessions.
- BPR-MF[6]: It is one of the commonly used matrix factorization methods, but cannot directly apply to session-based recommendations for the new session do not have feature vectors precomputed. Instead, we use the average value of item feature vectors that had occurred in the session before the predicting point, as the user feature vector (Rendle et al. 2009).
- Hierarchical RNN (HRNN)[7]: Proposed by Quadrana et al. (2017b), the model is a personalized RNN model with cross-session information transfer in a seamless way. HRNN relays end evolveds latent hidden states of the RNNs across user sessions.
- Neural Attentive Recommendation Machine (NARM)[8]: The model incorporates an item-level attention mechanism into RNN for capturing both the user's sequential behavior and main purpose in the current session (Li et al. 2017).

[6] https://github.com/bbc/theano-bpr.

[7] https://github.com/mquad/hgru4rec.

[8] https://github.com/lijingsdu/sessionRec_NARM.

4.4 Parameter settings

For user long-term interest modelling, we resort to the standard perplexity (Blei et al. 2003) and choose the topic number that leads to small perplexity and fast convergence. Therefore, we obtain the topic numbers $N_u^A = 70$, $N_u^L = 100$ and $N_u^W = 100$ for Adressa, Last.fm and Weibo-Net-Tweet, respectively. The embedding dimension D_e is set to 300. For time decay rate λ, we set it to 0.2 for Adressa dataset, but a relatively slow decay $\lambda = 0.1$ for the other two datasets. The sliding window size w is set as 150 in our experiments for the simplicity, which means we adopt 150 neighbourhood events of other users in public behavior mining procedure. In model training phase, the trad-off parameter λ_1 is set to 0.4 by grid-search over $\{0.2, 0.4, 0.6, 0.8\}$ and cross validation. The coefficient λ_2 of weight decayterm is set to $1e - 4$. We leverage stochastic gradient descent to optimize our model, and the learning rate is set to 0.001. Besides, we adopt one GRU layer with 100 hidden units in our model. The model is defined and trained in Theano.

5 Experiments

In this section, we evaluate the performances of our proposed models with four experiments. In the first experiment, we compare our DAINN model with state-of-the-art methods. The second experiment evaluates the influence of session length on the recommendation performance. In the third experiment, we evaluate the significance of different components of our model on recommendation performance. The last experiment explores the effectiveness of different recommendation algorithms in addressing cold-start issues.

5.1 Comparison against baselines

In this section, we present the experimental results of all baselines and our DAINN model with well-tuned parameters. To test the effectiveness of our proposed attenti- on mechanism with various side information, we concatenate the representations of day of week, hour of day and location, with high-dimensional representation learned from GRU in NARM model, to learn the final attention score α. The NARM with spatio-temporal enrichment is denoted as NARM + E. As shown in Table 3, it can be observed that our method can achieve superior performance than all the other baselines on all datasets. We also can obtain other observations: (1) the recommendation performance increase with the increasing number of k on all baselines. (2) all models perform better on Weibo and Adressa. It is mainly because the sparsity and unbalance characteristics appearing in Adressa dataset, and many meaningless words and noise can be found in Weibo dataset. (3) Among these competitors, Popular-based method get extremely bad results. The reason is that the method only provide user with random (if there are ties with items) or the same popular items, which fails to satisfy users' personalized demands. (4) The deep learning models including HRNN, NARM, NARM + E and DAINN, consistently outperform the other models on both datasets in terms of all evaluation metrics, despite the fact that Item KNN is a very competitive baseline. It is because the latter one cannot generalize the learned representations to new data. (5) Although HRNN considers the dynamics of user behaviors and achieves favorable results on datasets, it still does not adopt the contextual and semantic features and users' long-term interest. (6) NARM and NARM + E performs better than HRNN which can be attributed to the attention mechanism. However, compared with NARM model, NARM+E only improves

Table 3 Performance comparison of DAINN with baseline over three datasets

Method	POP	Item KNN	BPR-MF	HRNN	NARM	NARM + E	DAINN
Adressa							
Recall@5	0.011	0.118	0.112	0.141	0.197	0.213	0.253
Precision@5	0.009	0.108	0.101	0.120	0.195	0.211	0.232
F1-score	0.010	0.113	0.106	0.130	0.196	0.212	0.242
MRR@5	0.002	0.039	0.035	0.051	0.066	0.073	0.112
Recall@10	0.014	0.141	0.133	0.164	0.228	0.241	0.269
Precision@10	0.012	0.121	0.112	0.138	0.212	0.226	0.251
F1-score	0.013	0.131	0.122	0.149	0.220	0.233	0.260
MRR@10	0.003	0.051	0.042	0.064	0.078	0.086	0.125
Recall@20	0.021	0.162	0.156	0.187	0.253	0.265	0.287
Precision@20	0.016	0.134	0.127	0.156	0.232	0.249	0.268
F1-score	0.018	0.147	0.140	0.170	0.242	0.257	0.277
MRR@20	0.005	0.063	0.054	0.078	0.093	0.101	0.136
Last.fm							
Recall@5	0.106	0.201	0.189	0.231	0.303	0.316	0.347
Precision@5	0.089	0.194	0.187	0.221	0.277	0.285	0.334
F1-score	0.097	0.197	0.188	0.226	0.289	0.299	0.340
MRR@5	0.081	0.159	0.137	0.174	0.182	0.180*	0.228
Recall@10	0.118	0.219	0.203	0.256	0.320	0.317*	0.371
Precision@10	0.103	0.216	0.211	0.246	0.295	0.291*	0.359
F1-score	0.110	0.217	0.207	0.251	0.307	0.303*	0.365
MRR@10	0.095	0.173	0.151	0.183	0.194	0.192*	0.242
Recall@20	0.126	0.241	0.225	0.273	0.342	0.348	0.389
Precision@20	0.119	0.231	0.228	0.263	0.317	0.325	0.376
F1-score	0.122	0.236	0.226	0.268	0.329	0.336	0.382
MRR@20	0.107	0.186	0.162	0.194	0.201	0.208	0.253
Weibo-Net-Tweet							
Recall@5	0.087	0.211	0.195	0.216	0.289	0.283*	0.334
Precision@5	0.037	0.151	0.128	0.183	0.249	0.246*	0.304
F1-score	0.052	0.176	0.155	0.198	0.268	0.263*	0.318
MRR@5	0.062	0.141	0.113	0.152	0.163	0.160*	0.181
Recall@10	0.096	0.218	0.207	0.223	0.301	0.308	0.349
Precision@10	0.053	0.168	0.146	0.197	0.265	0.273	0.320
F1-score	0.068	0.190	0.171	0.209	0.282	0.289	0.334
MRR@10	0.075	0.149	0.127	0.158	0.176	0.183	0.194
Recall@20	0.103	0.224	0.215	0.236	0.314	0.326	0.357
Precision@20	0.071	0.184	0.163	0.215	0.282	0.295	0.336
F1-score	0.084	0.202	0.185	0.225	0.297	0.310	0.346
MRR@20	0.083	0.156	0.141	0.169	0.184	0.191	0.202

The numbers with asterisk represent the results of NARM + E are worse than NARM in our experiments

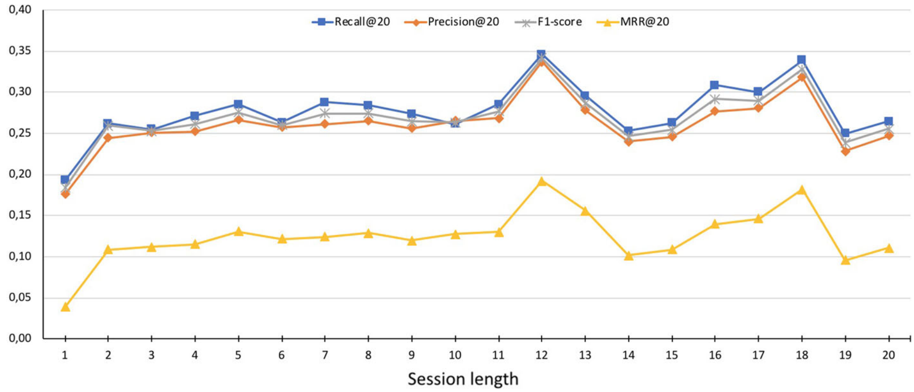


Fig. 4 The performance among different session lengths on Adressa dataset

little in Adressa dataset and appears unstable performance in Last.fm and Weibo dataset. We argue that this is because our attention mechanism performs better on experimental datasets and side information need to be integrated and modelled properly. Otherwise, it may cause counterprod- uctive effect. Besides, similar as HRNN, NARM and NARM+E do not consider users' historical records and relationship between sessions with other users. As a result, our proposed method outperforms NARM + E by (2.0%, 4.6%, 3.6%) with F1-score (k = 20) and (3.5%, 4.5%, 1.1%) with MRR@20 on Adressa, Last.fm and Weibo-Net-Tweet datasets, which also validates the effectiveness of the joint user long-term interest embedding and neighbourhood session embedding.

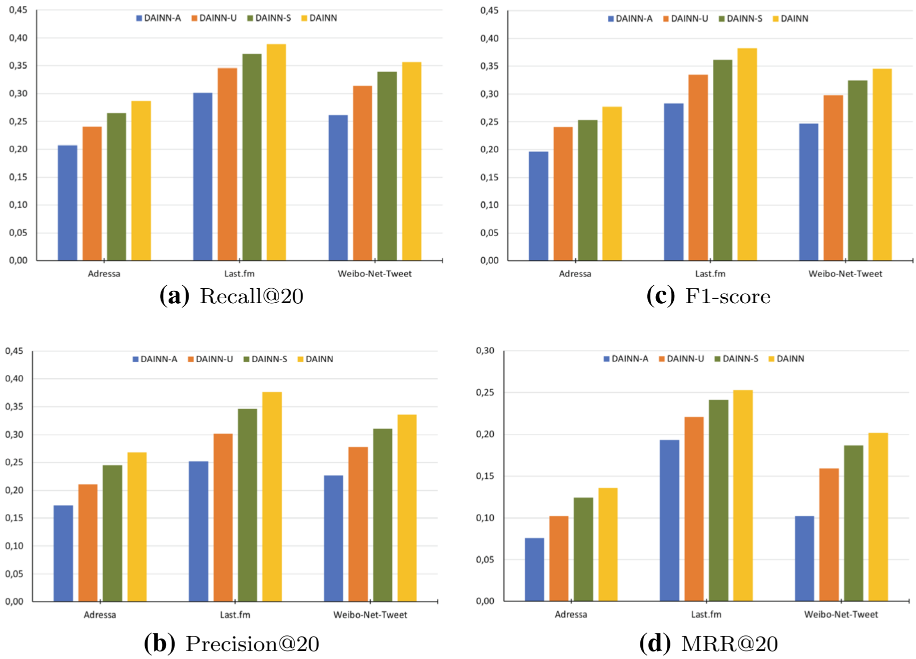
5.2 Evaluation on different session lengths

In this section, we study the impact of different session lengths on the recommendation performance. The attention scheme in our framework is based on the assumption that when a user browsing online, his/her click/play/post behavior frequently revolves his/her main purpose in the current session. However, if the user only clicks a few items, we can hardly capture the user's main purpose. Besides, we also need to make sure that the longer length of session, the better recommendation performance we can achieve for our DAINN model.

The experimental results on Adressa dataset are shown in Fig. 4. We can learn that: (1) In general, the recommendation performance of our model increases with the increasing number of session length, which indicates that DAINN model can capture user's main purpose more accurately on relatively long sessions. In other words, it needs a process to learn from the existing sequential behaviour features to make a better prediction. (2) However, when the session is too long, namely more than 18 in our experiments, the recommendation accuracy will decline. The reason we consider is that long session will bring more noise so that it increase the uncertainty and randomness of the user's behavior in the current session, which is to say that the user is very likely to click some items aimlessly, and thus it is hard for DAINN to capture the user's main purpose in the current session in this case.

5.3 Model component analysis

From Fig. 2, we can see that our method has four essential components including session-based public behavior mining in part (a), item semantic embedding in part (b), user long-



(a) Recall@20

(c) F1-score

(b) Precision@20

(d) MRR@20

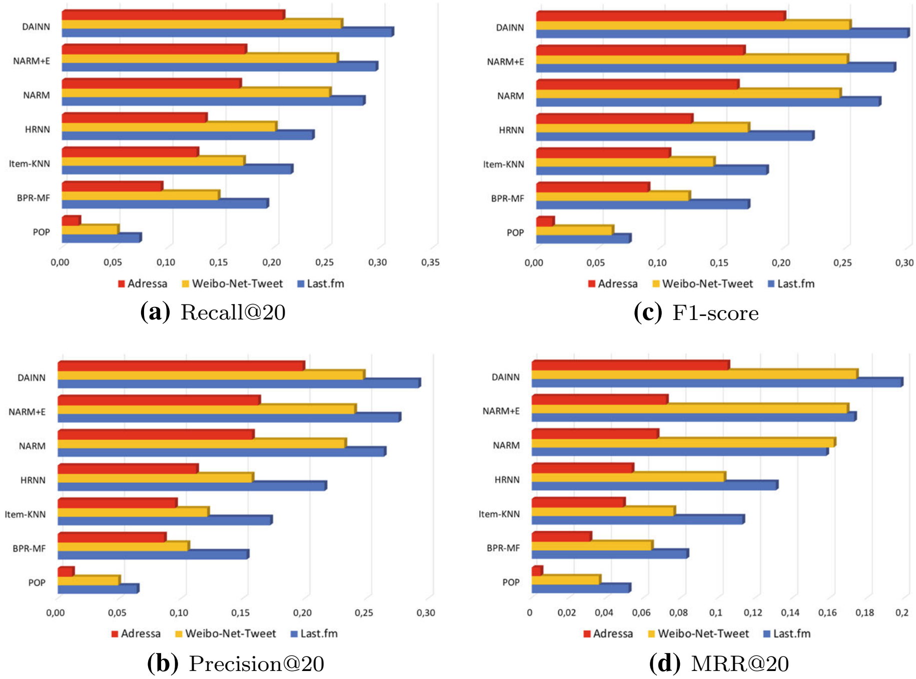
Fig. 5 Recommendation performance with different components of DAINN model

term interest modelling in part (c) and attention network. To verify the contribution of each component, we implement four variants of our approach: DAINN-S, DAINN-A, DAINN-U represent DAINN model without session-based public behavior mining, attention network and user long-term interest modelling in our framework. We cannot abandon item semantic embedding part since it is the base for other part.

The comparison results are shown in Fig. 5. The results show all the components contribute more or less to the final recommendation performance. Several observation can be found: (1) DAINN-A method results in inferior performance with, for instance F1-Score 19.6%, 28.3% and 24.7% of Adressa, Last.fm and Weibo datasets respectively. The results show that the attention scheme can capture the users' main purpose within and across session, and it can adaptively and smartly takes previous knowledge into consideration to capture the users' preference. Besides, the successful utilization of three key factors within attention network brings advantages when recommending items. (2) User long-term interest modelling is also essential in our framework. Compared with DAINN-U model, DAINN model get, for instance 3.6% promotion in terms of F1-score on Adressa dataset. The results demonstrate that users long-term interest modelling can capture users' broad interests and improves the recommendation performance significantly. (3) Our model also can benefit from session-based public behavior mining for DAINN gets a promotion from DAINN-S.

5.4 Cold-start problem

Additionally, we also conducted experiments to study the effectiveness of different recommendation algorithms in addressing cold-start issues on three kinds of datasets. As preprocessing, we removed users who have less than 5 events during sessions and more than 2 sessions in training sets. Beside, we also filtered users in testing set who were not



(a) Recall@20

(c) F1-score

(b) Precision@20

(d) MRR@20

Fig. 6 Recommendation for cold-start users

contained in training sets. Then, we randomly select 10,000 users among them to conduct the experiments.

The experimental results are shown in Fig. 6, from which we have the following observations: (1) our proposed DAINN model and NARM, NARM + E model still performs better consistently than other methods in recommending cold-start cases, which verifies the effectiveness of attention scheme used in session-based recommendation scenario; (2) by comparing the recommendation results in Table 3, the evaluation metrics of nearly all recommendation algorithms decreases, to different degrees, except Popular-based method. It is because the latter two methods consider less historical events than other methods. The recommendation performance of our DAINN model deteriorates more quickly than NARM and NARM + E, which is because the lack of user long-term interests makes DAINN consider only short-term interests of users when recommending items. (3) The recommendation performance of Item-KNN and BPR-MF drop drastically, which from another aspect proves the ineffectiveness of collaborative filtering methods in handling cold-start cases. (4) Our DAINN model still performs better than NARM and NARM + E model especially on Adressa and Last.fm datasets, since DAINN also considers different factors in attention scheme properly: hour of day, day of week and location. Besides, neighbourhood session information also brings positive influence for recommendation tasks.

6 Conclusion

In this paper, a novel dynamic attention-integrated neural network (DAINN) is proposed to address the problem of personalized session-based recommendation. In order to capture users' interests, we consider item semantic embedding, user long-term interest modelling and session-based public behavior mining in a unified framework, which can be trained end-to-end. By incorporating an attention mechanism into DAINN, our proposed approach can deal with the diverse variance of users' clicking behavior and capture the users' main purpose in the current session. DAINN can effectively learn users' real-time preference and conduct personalized recommendation. Evaluation on three different real-world datasets demonstrated the effectiveness of the proposed approach.

In the future, we will integrate other modal information, such as article image information, for recommendation. Meanwhile, both the nearest neighbour sessions and the importance of different neighbours should give new insights. Finally, we plan to investigate personalized session-based models in other domains, such as e-commerce and online advertisement.

Acknowledgements This work is partially funded by the Research Council of Norway (No. 245469).

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, *17*(6), 734–749.

Agarwal, D., & Chen, B.-C. (2009). Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 19–28). ACM.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. ArXiv preprint arXiv:1409.0473.

Bamshad, M., Dai, H., Tao, L., & Nakagawa, M. (2002). Using sequential and non-sequential patterns in predictive web usage mining tasks. In *Proceedings of the IEEE international conference on data mining (ICDM '02)*. IEEE Computer Society, Washington, DC, USA (pp. 669–672).

Baur, D., Buttgen, J., & Butz, A. (2012). Listening factors: A large-scale principal components analysis of long-term music listening histories. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1273–1276). ACM.

Blei, D. M., & Lafferty, J. D. (2006). Dynamic topic models. In *Proceedings of the 23rd international conference on machine learning* (pp. 113–120). ACM.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Chen, S., Moore, J. L., Turnbull, D., & Joachims, T. (2012). Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 714–722). ACM.

Chen, X., Xu, H., Zhang, Y., Tang, J., Cao, Y., Qin, Z., & Zha, H. (2018). Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 108–116). ACM.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. ArXiv preprint arXiv:1409.1259.

Cui, P., Wang, Z., & Su, Z. (2014). What videos are similar with you? Learning a common attributed representation for video recommendation. In *Proceedings of the 22nd ACM international conference on multimedia* (pp. 597–606). ACM.

Das, A. S., Mayur, D., Ashutosh, G., & Shyam, R. (2007). Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on world wide web* (pp. 271–280). ACM.

Ding, Y., & Li, X. (2005). Time weight collaborative filtering. In *Proceedings of the 14th ACM international conference on information and knowledge management* (pp. 485–492). ACM.

Gopalan, P. K., Charlin, L., & Blei, D. (2014). Content-based recommendations with Poisson factorization. In *Advances in neural information processing systems* (pp. 3176–3184).

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 6645–6649). IEEE.

Greene, D., & Cross, J. P. (2016). Exploring the political agenda of the European parliament using a dynamic topic modelling approach. ArXiv preprint arXiv:1607.03055.

Grefenstette, E., Hermann, K. M., Suleyman, M., & Blunsom, P. (2015). Learning to transduce with unbounded memory. In *Advances in neural information processing systems* (pp. 1828–1836).

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. ArXiv preprint arXiv:1511.06939.

Hidasi, B., Quadrana, M., Karatzoglou, A., & Tikk, D. (2016). Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM conference on recommender systems* (pp. 241–248). ACM.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 89–115.

Hong, J., & Fang, M. (2015). Sentiment analysis with deeply learned distributed representations of variable length texts. Technical report, Stanford University.

Huang, J., Zhao, W. X., Dou, H., Wen, J.-R., & Chang, E. Y. (2018). Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st international ACM SIGIR conference on research and development in information retrieval* (pp. 505–514). ACM.

Jing, Z., Liu, B., Tang, J., Chen, T., & Li, J. (2013). Social influence locality for modelling retweeting behaviors. *IJCAI*, *13*, 2761–2767.

Jurafsky, D., & Martin, J. H. (2014). *Speech and language processing* (Vol. 3). London: Pearson.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1746–1751).

Koren, Y., Robert, B., & Chris, V. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., & Socher, R. (2016). Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning* (pp. 1378–1387).

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. *AAAI*, *333*, 2267–2273.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on world wide web* (pp. 661–670). ACM.

Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., & Ma, J. (2017). Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 1419–1428). ACM.

Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. *AAAI/IAAI*, *23*, 187–192.

Pan, Y., Mei, T., Yao, T., Li, H., & Rui, Y. (2016). Jointly modelling embedding and translation to bridge video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4594–4602).

Pei, W., Yang, J., Sun, Z., Zhang, J., Bozzon, A., & Tax, D. M. (2017) Interacting attention-gated recurrent networks for recommendation. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 1459–1468). ACM.

Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, *2*(1), 37–63.

Quadrana, M., Karatzoglou, A., Hidasi, B., & Cremonesi, P. (2017). Personalizing session-based recommendations with hierarchical recurrent neural networks. ArXiv preprint arXiv:1706.04148.

Rao, J., Jia, A., Feng, Y., & Zhao, D. (2013). Personalized news recommendation using ontologies harvested from the web. In *International conference on web-age information management* (pp. 781–787). Berlin: Springer.

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

Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on machine learning* (pp. 791–798). ACM.

Sarwar, B., Karypis, G., Konstan, J., & Riedl J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on world wide web* (pp. 285–295). ACM.

Schafer, J. B., Konstan, J., & Riedl, J. (1999). Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on electronic commerce* (pp. 158–166). ACM.

Schedl, M. (2016). The LFM-1b dataset for music retrieval and recommendation. In *Proceedings of the ACM international conference on multimedia retrieval (ICMR 2016)*, New York, USA.

Shani, G., Heckerman, D., & Brafman, R. I. (2005). An MDP-based recommender system. *Journal of Machine Learning Research*, *6*, 1265–1295.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631–1642).

Song, Y., Elkahky, A. M., & He, X. (2016). Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval* (pp. 909–912). ACM.

Sukhbaatar, S., Weston, J., & Fergus, R. (2015). End-to-end memory networks. In *Advances in neural information processing systems* (pp. 2440–2448).

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, *2009*, 2–5.

Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. ArXiv preprint arXiv:1503.00075.

Tan, Y. K., Xu, X., & Liu, Y. (2016). Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems* (pp. 17–22). ACM.

Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In *Advances in neural information processing systems* (pp. 2643–2651).

Vinyals, O., Kaiser, A., Koo, T., Petrov, S., Sutskever, I., & Hinton, G. (2015). Grammar as a foreign language. In *Advances in neural information processing systems* (pp. 2773–2781).

Voorhees, E. M. (1999). The TREC-8 question answering track report. In *TREC* (Vol. 99, pp. 77–82).

Wang, C., & Blei, D. M. (2011). Collaborative topic modelling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 448–456). ACM.

Wang, H., Wang, N., & Yeung, D.-Y. (2015). Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1235–1244). ACM.

Wang, H., Zhang, F., Xie, X., & Guo, M. (2018). DKN: Deep knowledge-aware network for news recommendation. ArXiv preprint arXiv:1801.08284.

Weimer, M., Karatzoglou, A., Le, Q. V., & Smola, A. J. (2008). Cofi rank-maximum margin matrix factorization for collaborative ranking. In *Advances in neural information processing systems* (pp. 1593–1600).

Wu, C., & Yan, M. (2017). Session-aware information embedding for E-commerce product recommendation. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 2379–2382). ACM.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048–2057).

Yang, L., Ai, Q., Guo, J., & Croft, W. B. (2016). aNMM: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM international on conference on information and knowledge management* (pp. 287–296). ACM.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 1480–1489).

Yap, G.-E., Li, X.-L., & Philip, S. Y. (2012). Effective next-items recommendation via personalized sequential pattern mining. In *International conference on database systems for advanced applications* (pp. 48–64). Berlin: Springer.

Yu, F., Liu, Q., Wu, S., Wang, L., & Tan, T. (2016). A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval* (pp. 729–732). ACM.

Zhang, Q., Gong, Y., Wu, J., Huang, H., & Huang, X. (2016). Retweet prediction with attention-based deep neural network. In *Proceedings of the 25th ACM international on conference on information and knowledge management* (pp. 75–84). ACM.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems* (pp. 649–657).

Zheleva, E., Guiver, J., Mendes Rodrigues, E., & Milić-Frayling, N. (2010). Statistical models of music-listening sessions in social media. In *Proceedings of the 19th international conference on world wide web* (pp. 1019–1028). ACM.

Zimdars, A., Chickeringm, D. M.,& Meek, C. (2001). Using temporal data for making recommendations. In *Proceedings of the seventeenth conference on uncertainty in artificial intelligence* (pp. 580–588). Burlington: Morgan Kaufmann Publishers Inc.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.