Time-Aware Attentive Neural Network for News Recommendation with Long- and Short-Term User Representation

Yitong Pang[1], Yiming Zhang[1], Jianing Tong[2], and Zhihua Wei[1](✉)

[1] Department of Computer Science and Technology,
Tongji University, Shanghai, China
{1930796,1652325,zhihua_wei}@tongji.edu.cn
[2] DellEmc, Shanghai, China
johnny_tong@dell.com

Abstract. News recommendation is very critical to help users quickly find news satisfying their preferences. Modeling user interests with accurate user representations is a challenging task in news recommendation. Existing methods usually utilize recurrent neural networks to capture the short-term user interests, and have achieved promising performance. However, existing methods ignore the user interest drifts caused by time interval in the short session. Thus they always assume the short-term user interests are stable, which might lead to suboptimal performance. To address this issue, we propose the novel model named Time-aware Attentive Neural Network with Long-term and Short-term User Representation (TANN). Specifically, to reduce the influence of interest drifts, we propose the Time-aware Self-Attention (T-SA) which considers the time interval information about user browsing history. We learn the short-term user representations from their recently browsing news through the T-SA. In addition, we learn more informative news representations from the historical readers and the contents of news articles. Moreover, we adopt the latent factor model to build the long-term user representations from the entire browsing history. We combine the short-term and long-term user representations to capture more accurate user interests. Extensive experiments on two public datasets show that our model outperforms several state-of-the-art methods.

Keywords: News recommendation · Self attention · Time-aware · Long-term interest · Short-time interest · Representation learning

1 Introduction

Nowadays, online news platforms have become popular with people to acquire daily information, such as MSN News and Google News. However, with the explosion of news contents and services, users are overwhelmed by tremendous news. News recommendation can find news that satisfies the personalized interests of

G. Li et al. (Eds.): KSEM 2020, LNAI 12275, pp. 76–87, 2020.
https://doi.org/10.1007/978-3-030-55393-7_7

users, and is an important method to alleviate the information overload [7]. Therefore, news recommendation has received the increasing attention on both academics fields and industry fields.

In news recommendation, the key task is learning the accurate user representations to reflect the user interests. Because of the uncertain user behavior and limited information, it is difficult to capture appropriate user interests. Traditional methods, like collaborative-filtering (CF) based methods ignore the sequence information about user browsing history. They can not learn the current interest of users exactly. Recently, some novel models based on deep learning were proposed for personalized news recommendation. Some deep learning based methods utilize the recurrent neural networks (RNN) and attention mechanism to capture the short-term user interests from recently viewed news [14,20,22,23]. Besides, some methods exploit to utilize the long-term and short-term interest of users together for more accurate representation [1,7]. For example, [1] proposed to learn long-term interests from user IDs and capture short-term user interests from recently viewed news by GRU model.

Although these deep learning based methods have achieved encouraging results, they still have two potential limitations. Firstly, they ignore the short-term user interest drifts, and consider that user preferences are stable in the short term. Short-term user interest is usually dynamic, but it is critical for making the recommendation decisions. Short-term user interests usually are dynamic but important to making news recommendation decisions. For example, the current interest of a user may have changed when he resumes browsing news after a short interval. Secondly, they generally recommend news by matching user interests with news content, but overlook the effective collaborative information. It is worth noting that users with similar interests may also read similar news, e.g., if Peter has similar browsing history to Bob and has browsed the news C, then Bob will probably also read C, even if the content of C is different from the news that Bob recently browsed. These mentioned methods fail to take into account this collaborative information.

For addressing above issues, this paper proposes a **T**ime-aware **A**ttentive **N**eural **N**etwork with long- and short-term user representation for news recommendation (TANN). In order to reduce the influence of user interest drifts, we propose the time-aware self-attention (T-SA) which considers the time interval information between two browsing records. We learn the short-term representations of users from recently browsed news via the T-SA. In addition, to integrate the collaborative information, we apply attention mechanism to learn informative news representations from the historical readers and the contents of news articles. Furthermore, we utilize latent factor model to model the long-term interests from entire browsing history of users, and combine the long-term interests with the short-term user interests for better representations. The experimental results on two real-world datasets show our method has achieved the promising performance and is superior to several state-of-the-art methods.

2 Related Work

Today, personalized news recommendations have received extensive attention from academia and industry, and have been extensively studied. A variety of news recommendation methods have been proposed, including conventional methods and methods based on deep learning [18,23].

Conventional methods include the CF-based methods, content-based methods and hybrid methods. CF-based methods assume that users with similar behavior will show similar preferences for news [3,6,7]. They usually learn the latent representations from entire historical interactions to get the general user interests. For example, latent factor model [9] directly models the user-item interaction with inner product. But CF-based methods have to face the problem of cold start and lack the ability to capture the short-term interests of users.

Content-based and hybrid recommendation can alleviate the cold-start problem. Content-based methods analyze the actual content or attributes of the news articles for recommendation Content-based methods analyze the content of historical news articles browsed by users for recommendation [12,16,17]. For example, [16] adopt vector space model similarity to evaluate the relevance of different news and recommend relevant news to users. Hybrid methods usually combine several different methods [10,11,13], such as SCENE [10] proposed a two-stage recommendation framework to blend the CF-based method and content-based method. However, these methods neglect the sequence information in user browsing history and can not effectively learn the short-term user interests. In addition, they fail to consider the contexts and semantic information in news which are important to learning representations of news and users.

Deep learning based methods are proposed for personalized news recommendation recently. These methods utilize the neural network models such as recurrent neural network and attention mechanisms, to learn short-term user interests from the sequence of the recently browsed news. Besides, these methods learn deep news features from news contents via neural networks [1,7,14,18,21–23]. For example, [14] adopt denoising auto-encoder to obtain effective news feature from news attributes, and learn users representations from browsed news via GRU network. Moreover, some works exploit to integrate the long-term and short-term interest of users for more accurate representations [1,7,24], such as [24] proposed to the time-LSTM which uses the time gate to control the impact of time, and introduces time interval information to obtain more reasonable user representations.

However, these deep learning based methods ignore the short-term user interest drifts and assume that user interests are stable in the short term. In addition, they usually take the idea of content-based methods, but ignore the effective collaborative information. Inspired by [2,24], we propose a time-aware attentive neural network, and consider the time interval information to alleviate the effects of user interest drifts. Furthermore, we exploit to learn the news representations from the content and the historical reader of news to integrate the collaborative information.

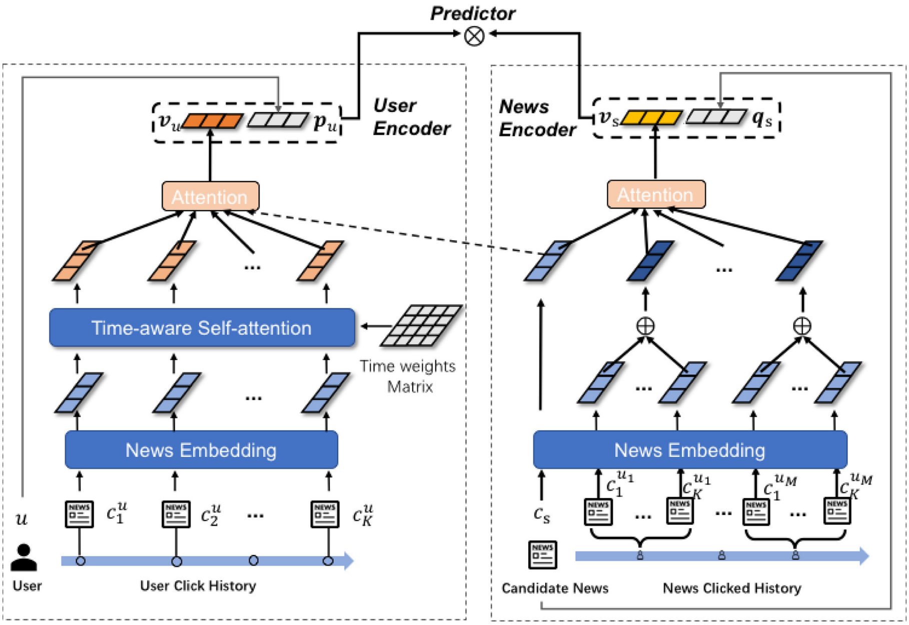

Fig. 1. The architecture of the proposed *TANN* for news recommendation.

3 Methodology

In this part, we present the TANN for news recommendation. The architecture of our proposed model is shown in Fig. 1. There are three major modules, i.e., the *User Encoder* with time-aware self-attention and latent factor model, the *News Encoder* with attention mechanism, and the *Predictor* to predict the probability that users will click on candidate news articles.

3.1 Problem Formulation

Assume that there are m users $U = \{u_1, u_2, ..., u_m\}$ and n news articles $V = \{v_1, v_2, ..., v_n\}$. According to the implicit feedback of users, the entire browsing data can be defined as matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, where $\mathbf{M}_{u,s} = 1$ indicates the user u browsed the news s, otherwise $\mathbf{M}_{u,s} = 0$. For each user, we chronologically organize the browsing history as a sequence of tuples $O^u = \{(v_j^u, t_j^u)\}_{j=1}^{l_u}$, where $t_1^u \leq t_2^u \leq ... \leq t_{l_u}^u$, v_j^u is the news in V, l_u indicates the number of articles viewed by user u, and (v_j^u, t_j^u) means that user u browsed news v_s^u at time t_s^u. In this paper, we exploit to build a prediction function $\hat{y}_{u,s} = F(u, s; \Theta) \in [0, 1]$ from the user-item matrix $\mathbf{M}$ and the click sequences O^u. Prediction function $F(u, s; \Theta)$ is used to calculate the probability that user u will click on a candidate news s.

3.2 User Encoder

In our model, we learn user representations from the browsing history via the *User Encoder*. It consists of two modules, namely, the long-term user representation module for capturing user consistent preferences and a short-term representation part for modeling temporal interests.

Long-Term Interest Representation. Users usually have long-term interests in reading news. The long-term user interest is stable and will not change much over time. For example, a football fan might browse many sports news about La Liga for years. To distill user consistent preferences, we exploit to learn long-term user representations from the user browsing history. Specifically, we learn the long-term representations from the entire user-item interaction data via latent factor model [9]:

$$\mathbf{M} = \mathbf{W}_U \cdot \mathbf{W}_V^T \tag{1}$$

where $\mathbf{W}_U \in \mathbb{R}^{m \times k}$ and $\mathbf{W}_V \in \mathbb{R}^{n \times k}$ are the user feature matrix and item feature matrix, respectively. We denote $\mathbf{p}_u = \mathbf{W}_U[u] \in \mathbb{R}^k$ as the long-term user representation of u. Besides, $\mathbf{q}_s = \mathbf{W}_V[s] \in \mathbb{R}^k$ is the embedding of news s, and k is the dimension of the latent space.

Short-Term Interest Representation. Short-term user interest reflects a common situation: users tend to be attracted to something temporarily. For example, the temporal interest of a user may have changed when he resumes browsing news after a short interval. To reduce the influence of interest drifts, we propose **T**ime-aware **S**elf-**A**ttention (T-SA) mechanism to learn the short-term user interests. T-SA considers two aspects of time factor: time order information and time interval information. Given the recent browsing history of user u: $C_u = \{(v_1, t_1), (v_2, t_2), ..., (v_K, t_K)\} \subseteq O^u$, where $t_i < t_{i+1}$, and the corresponding news embedding sequence $\{\mathbf{e}_1, ..., \mathbf{e}_K\}$ through the News Embedding Layer. We adopt the pre-trained word emebedding model to obtain the news embeddings.

For the time order information, we encode it into the news embedding. The dimension size of the news embedding is h, and the calculation of each dimension in the time order vector is as follows:

$$PE(pos, 2i) = sin(pos/10000^{2i/h}) \tag{2}$$

$$PE(pos, 2i+1) = cos(pos/10000^{(2i+1)/h}) \tag{3}$$

where $pos \in 1, 2, ..., K$ is the time order. The time order code is PE_{pos}, and the new vector $\_\mathbf{e}_i$ which contains time order information as follows:

$$\_\mathbf{e}_i = \mathbf{e}_i + PE_i \tag{4}$$

For the time interval information, we obtain the time interval sequence $T_\varDelta = \{\varDelta_1, \varDelta_2, ...\varDelta_K\}$, where $\varDelta_i = t_i - t_{i-1}$ and $\varDelta_1 = 0$. Based on the attention mechanism, we propose a time weight matrix to control the attention weight between two news embeddings. Through the time weight, we can achieve that the bigger the time interval is, the smaller the attention weights between the two news are. Each embedding attention weight in the sequence is computed as follows:

$$\alpha_{i,j} = \frac{w_{i,j}^{\Delta} exp(\_\mathbf{e}_i^T \mathbf{Q}^w \_\mathbf{e}_j)}{\sum_{k=1}^{K} w_{i,k}^{\Delta} exp(\_\mathbf{e}_i^T \mathbf{Q}^w \_\mathbf{e}_k)} \tag{5}$$

$$\mathbf{e}_i = \sum_{k=1}^{K} \alpha_{i,k} \_\mathbf{e}_k \tag{6}$$

where $\mathbf{Q}^w \in \mathbb{R}^{h*h}$ is the trainable projection parameter. $w_{i,j}^{\Delta}$ is the time interval weight between i-th news and j-th news. The principle of time weight setting is as follows:

$$w_{i,i}^{\Delta} = 1 \tag{7}$$

$$w_{i,j}^{\Delta} = w_{j,i}^{\Delta} = w_{i,j-1}^{\Delta} * p_{j-1,j}, i < j \tag{8}$$

$$p_{j-1,j} = \begin{cases} q < 1, \, if \Delta_j \geq \Delta_{threshold} \\ 1, \quad\quad otherwise \end{cases} \tag{9}$$

where $\Delta_{threshold}$ is the time threshold and $q \in (0, 1)$ control the magnitude of the change in time weight. For example, the time interval sequence is $\{0, 100, 20, 50\}$, and the $\Delta_{threshold} = 30$. Then the time interval weight matrix is shown as follows:

$$\begin{pmatrix} 1, \; q, \; q, \; q^2 \\ q, \; 1, \; 1, \; q \\ q, \; 1, \; 1, \; q \\ q^2, \; q, \; q, \; 1 \end{pmatrix} \tag{10}$$

We can get a new news embedding sequence $\{\mathbf{e}_1', \mathbf{e}_2', ..., \mathbf{e}_K'\}$ through the T-SA. This sequence considers the influence of time factor on short-term user interest. T-SA can learn the sequence correlation of each news article, and detect weak sequence correlation because of taking into account the time interval. Therefore, T-SA can alleviate the effects of short-term user interest drifts.

Moreover, we apply the attention mechanism to learn relevance between the news recently clicked by the user and the candidate news v_s. The specific calculation method is as follows:

$$\mathbf{d}_i = \tanh(\mathbf{W}_u \mathbf{e}_i' + \mathbf{w}_u) \tag{11}$$

$$\mathbf{d}^s = \tanh(\mathbf{W}_u \mathbf{e}_s + \mathbf{w}_u) \tag{12}$$

$$\alpha_i^u = \frac{exp(\mathbf{q}^u(\mathbf{d}_i + \mathbf{d}^s))}{\sum_{j=1}^{K} exp(\mathbf{q}^u(\mathbf{d}_j + \mathbf{d}^s))} \tag{13}$$

$$\mathbf{v}_u = \sum_{i=1}^{K} \alpha_i^u \mathbf{e}_i' \tag{14}$$

where $\mathbf{e}_s$ is the embedding of candidate news, $\mathbf{W}_u \in \mathbb{R}^{g*h}, \mathbf{q}^u \in \mathbb{R}^g$ are trainable parameters, and $\mathbf{w}_u \in \mathbb{R}^g$ is bias parameter. $\mathbf{v}_u$ denotes the short term interest representation of user u.

3.3 News Encoder

The *News Encoder* is designed to extract the abstract features of candidate news. We can learn the collaborative feature from the historical readers information. To leverage the collaborative information, we learn representations of a news article from the article content and users who recently browsed this article. For a candidate news v_s, $U_s = \{u_1, u_2, ..., u_M\}$ denotes the users who have recently browsed v_s. $C_i = \{v_1^{u_i}, v_2^{u_i}, ..., v_K^{u_i}\}$ denotes the news recently browsed by user u_i. As mentioned above, We can obtain the embedding of each news through the News Embedding Layer. The embedding set corresponding to the C_i is $E_i = \{\mathbf{e}_1^{u_i}, \mathbf{e}_2^{u_i}, ..., \mathbf{e}_K^{u_i}\}$. For the set of users who have clicked v_s, we apply the average sum of news embedding to get the content embedding of each user:

$$\mathbf{e}^{u_k} = \sum_{i=1}^{K} \frac{\mathbf{e}_i^{u_k}}{K} \tag{15}$$

where $\mathbf{e}_i^{u_k} \in \mathbb{R}^h$ represents the vector of the i-th news that user u_k has browsed. The set $E_s^u = \{\mathbf{e}^{u_1}, \mathbf{e}^{u_2}, ..., \mathbf{e}^{u_M}\}$ represents the textual feature of the news recently clicked by the each user. It indicates the reading features of current readers of news v_s and indirectly reflects the context of news v_s. For the each embedding $\mathbf{e}_i^{s'}$ in the sequence $\{\mathbf{e}_s, \mathbf{e}^{u_1}, ..., \mathbf{e}^{u_M}\}$, we utilize the attention mechanism as follows:

$$\alpha_i^s = \frac{exp(\mathbf{q}^w \tanh(\mathbf{W}_c \mathbf{e}_i^{s'} + \mathbf{w}_b))}{\sum_{j=1}^{M+1} exp(\mathbf{q}^w \tanh(\mathbf{W}_c \mathbf{e}_j^{s'} + \mathbf{w}_b))} \tag{16}$$

where $\mathbf{W}_c \in \mathbb{R}^{g*h}, \mathbf{q}^w \in \mathbb{R}^g$ are trainable parameters, and $\mathbf{w}_b \in \mathbb{R}^g$ is bias parameter. α_i^s represents the weight of the i-th embedding in the sequence. The final news representation $\mathbf{v}_s$ which contains text information and collaborative information is denoted as follows:

$$\mathbf{v}_s = \sum_{i=1}^{M+1} \alpha_i^s \mathbf{e}_i^{s'} \tag{17}$$

Besides, for the candidate news v_s, we also get the latent news embedding $\mathbf{q}_s$ by latent factor model.

3.4 Predictor

To obtain more accurate representations, we combine the long- and short-term user representation together. The final user and news representation are as followed:

$$\mathbf{v}_u^U = [\mathbf{v}_u; \mathbf{p}_u] \tag{18}$$

Table 1. Datasets Statistics.

Dataset	#user	#news	#interaction	Avg. #articles seen per user
Globo	322,897	46,033	2,988,181	9.25
Adressa	640,503	20,428	2,817,881	4.40

$$\mathbf{v}_s^I = [\mathbf{v}_s; \mathbf{q}_s] \tag{19}$$

where $[\cdot; \cdot]$ is the concatenation operation.

To predict the probability of the user u clicking the candidate news article v_s, we utilize the cosine function as the *Predictor*: $\hat{r}_{u,s} = cosine(\mathbf{v}_u^U, \mathbf{v}_s^I)$. $\hat{r}_{u,s}$ denotes the clicking probability.

3.5 Loss Function

We apply the pairwise learning method to train our proposed model. For the input triple $< u, p, n >$, where u, p, n respectively denote users, positive sample and negative sample, we minimize objection function as follows:

$$\arg\min_{\Theta} \sum_{(u,p,n)\in D} max\{0, m - (\hat{r}_{u,p} - \hat{r}_{u,n})\} + \lambda\Omega(\Theta) \tag{20}$$

where m is the margin between positive and negative sample, Θ is the trainable parameters of the model, $\Omega(\cdot)$ denotes the L2 regularization and λ is the penalty weight.

Since not all users and news articles can participate in the model training, just like the new users or new articles, we cannot obtain the long-term representation of each user during the prediction phase of our model. In order to solve this problem, we use a random masking strategy followed [1] during model training. Specifically, we randomly mask the long-term user representations $\mathbf{p}_u$ and the latent representation of news $\mathbf{q}_s$ with a certain probability p_m. The mask operation sets all dimensions of the vector to zero. Thus, the representations can be reformulated as:

$$\begin{aligned} \mathbf{p}_u &= p \cdot \mathbf{W}_u[u] \\ \mathbf{q}_s &= p \cdot \mathbf{W}_s[s] \\ p &\sim M(1, 1 - p_m) \end{aligned} \tag{21}$$

where M denotes the binomial distribution This design is in line with the actual situation and can reduce the impact of cold start on the model prediction [1].

4 Experiments

4.1 Experiments Setup

Dataset and Evaluation Protocol. We conduct experiments on two public real-world datasets: Globo[1] and Adressa[2]. Data statistics is shown in Table 1.

[1] https://www.kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom.

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

Globo dataset comes from a popular news platform in Brazil, which provides the pre-trained content embeddings of news articles. Adressa dataset [4] comes from a social news platform in Norwegian. We use one-week version of Adressa and adopt the pre-trained Glove embeddings [15] to build the news embeddings.

For each user in each dataset, the interaction history is split into the training set, validation set and test set with the ratio of 80%, 10% and 10% respectively. We use negative sampling to construct training data, and treat news articles that have not been browsed by users as negative samples. During model training, the validation set is used to adjust the hyper-parameters. For each user in the test set, we sample 99 negative items and pair them with the positive sample. Then each model will calculate the score for each user-item interaction in the test set.

To evaluate the recommendation performance, we employ three widely-adopted metrics: *Hit Ratio* (HR), *Normalize Discounted Cumulative Gain* (NDCG) and *Area Under Curve* (AUC) [6].

Comparison Methods. We compare our model TANN with some recent state-of-the-art (SOTA) methods, including ConvNCF [6], DeepFM [5], NRMS [21], NAML [19], DAN [23], LSTUR-ini [1]. These models are based on deep learning methods.

Implementation Details. We implement TANN based on Tensorflow. We optimize the AUC on the validation set to obtain the optimal hyper-parameter settings. They are setting as follows: we set negative sampling size $S = 3$ with random strategy; we set the sequence length $K = 10$; Margin is set as $m = 0.05$; And Adam [8] is used to optimize the parameters with learning parameter of 0.0003. The mask probability p_m is 0.7. Besides we set the $\Delta_{threshold}$ to be 1000 seconds. Regularization penalty factor λ is set to 0.005 and the dropout rate is 0.5. The batch size is 1024. We train the models at most 100 epoches.

For a fair comparison, we employ the source codes of all the SOTA methods from Github and fine-tune parameters to get the best performance for these models according to their works.

4.2 Performance Comparison

Comparisons of Different Models. First of all, we conduct experiments to compare our model with the SOTA methods on two datasets. We show the detailed results on three different metrics in Table 2. We can obtain several key observations from Table 2:

Firstly, the performance of our method is significantly improved compared to all comparative methods. Especially, NDCG@5 increases by more than 6.41% and HR@20 increases by more than 3.34%. We believe that the superior performance of the model mainly stems from its three advantages: (1) TANN considers the time interval information when learning the short-term user interest. (2) Our method considers both the long-term user interest and the short-term interest for better user representation. (3) We introduce the information of news readers and

Table 2. Comparison of different methods on two Datasets for Top-K news recommendation.

Datasets	Methods	HR@K			NDCG@K			AUC
		$K = 5$	$K = 10$	$K = 20$	$K = 5$	$K = 10$	$K = 20$	
Globo.	ConvNCF	0.7439	0.8260	0.8587	0.5548	0.5819	0.5902	0.8612
	DeepFM	0.7559	0.8223	0.8398	0.5489	0.5701	0.5745	0.8779
	NRMS	0.7536	0.8284	0.8483	0.5610	0.6172	0.6728	0.9106
	DAN	0.7543	0.8356	0.8527	0.5317	0.5802	0.6825	0.9027
	NAML	0.8301	0.8370	0.8342	0.5950	0.6276	0.6774	0.9141
	LSTUR-ini	0.8635	0.9383	0.9483	0.6172	0.6728	0.6906	0.9206
	TANN	**0.8748**	**0.9428**	**0.9631**	**0.6568**	**0.6864**	**0.7091**	**0.9601**
Adressa.	ConvNCF	0.7107	0.7895	0.8021	0.4719	0.5368	0.5583	0.8550
	DeepFM	0.7457	0.8013	0.8101	0.5334	0.5549	0.5573	0.9259
	NRMS	0.7468	0.8142	0.8233	0.5210	0.5572	0.5728	0.9316
	DAN	0.7516	0.8217	0.8438	0.5110	0.5426	0.5701	0.9231
	NAML	0.7601	0.8311	0.8561	0.5486	0.5788	0.6289	0.9308
	LSTUR-ini	0.7935	0.8257	0.8783	0.5732	0.5923	0.6406	0.9362
	TANN	**0.8088**	**0.8392**	**0.9076**	**0.5886**	**0.6049**	**0.6499**	**0.9694**

add the collaborative signals to represent news, which can improve the accuracy of news recommendations.

Secondly, the methods that consider the reading sequences of users (e.g., DAN, NRMS) is better than the collaborative-filtering based method (e.g., ConvNCF). This is because CF-based methods cannot reflect the current interest of users for news recommendation.

Thirdly, the methods which exploit to combine the long-term and short-term user representations (e.g., LSTUR-ini and TANN) outperform other methods. This may be because these methods have stronger feature extraction capabilities and can model complex and varied user preferences from reading history. For the models using a single user representation, often cannot fully reflect the users interest.

Comparisons of TANN Variants. Moreover, to show the effectiveness of the design of our method, we compare among the variants of TANN with respect several aspects: Time-aware self attention, combining of long-term and short-term user interests, and the novel representation of news. The experimental results on the Globo dataset are shown in Table 3.

As can be seen that, there is dramatic decline in performance when removing the T-SA. This proves the effectiveness of our design. In the design of T-SA, we consider the time interval information and adopt the self-attention to learn the feature from the short reading sequences. Besides, removing long-term interest representation can also lead to poor experimental results. It demonstrates that considering both long-term and short-term user interests is necessary. Moreover, we remove the novel news representation and only use news article embedding as the news representation. The decreased performance shows the rationality of our news representation method.

Table 3. Comparison of TANN Variants.

Model	HR@K		NDCG@K	
	$K = 5$	$K = 10$	$K = 5$	$K = 10$
TANN without T-SA	0.8021	0.8319	0.5248	0.5619
TANN without long-term interest	0.8428	0.8735	0.5752	0.5876
TANN without novel news representation	0.8631	0.9143	0.6252	0.6376
TANN	**0.8748**	**0.9428**	**0.6568**	**0.6864**

5 Conclusion

In this paper, we propose a novel news recommendation model, called Time-aware Attentive Neural Network with long- and short-term user representation (TANN). We propose to take the time interval of user reading into account and design the novel self-attention model to learn more accurate user short-term interest. Besides, we propose to learn the news feature from the historical readers and contents of news articles via the attention mechanism. Furthermore, we use the latent factor model to learn the short-term interests and combine the long-term and short-term interests together for better representation. From extensive experimental results, it can be proved that TANN outperforms previous advanced methods in news recommendation performance. In the future, we plan to extend TANN to other recommendation scenarios.

Acknowledgements. This work is supported by the project of National Key research and development plan (Grant No. 213) and the National Natural Science Foundation of China (Grant No. 61976160, No. 61673301), and Major project of Ministry of Public Security (Grant No. 20170004).

References

1. An, M., Wu, F., Wu, C., Zhang, K., Liu, Z., Xie, X.: Neural news recommendation with long-and short-term user representations. In: Proceedings of the 57th ACL, pp. 336–345 (2019)
2. Chen, X., Zhang, Y., Qin, Z.: Dynamic explainable recommendation based on neural attentive models. In: Proceedings of the AAAI, vol. 33, pp. 53–60 (2019)
3. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proceedings of the 16th WWW, pp. 271–280. ACM (2007)
4. Gulla, J.A., Zhang, L., Liu, P., Özgöbek, Ö., Su, X.: The Adressa dataset for news recommendation. In: Proceedings of the International Conference on Web Intelligence, pp. 1042–1048. ACM (2017)
5. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: DeepFM: a factorization-machine based neural network for CTR prediction. arXiv preprint arXiv:1703.04247 (2017)

6. He, X., Du, X., Wang, X., Tian, F., Tang, J., Chua, T.S.: Outer product-based neural collaborative filtering. In: Proceedings of the 27th IJCAI, pp. 2227–2233. AAAI Press (2018)
7. Hu, L., Li, C., Shi, C., Yang, C., Shao, C.: Graph neural news recommendation with long-term and short-term interest modeling. Inf. Process. Manag. **57**(2), 102142 (2020)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
10. Li, L., Wang, D., Li, T., Knox, D., Padmanabhan, B.: Scene: a scalable two-stage personalized news recommendation system. In: Proceedings of the 34th ACM SIGIR, pp. 125–134 (2011)
11. Li, L., Zheng, L., Yang, F., Li, T.: Modeling and broadening temporal user interest in personalized news recommendation. Expert Syst. Appl. **41**(7), 3168–3177 (2014)
12. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th WWW, pp. 661–670 (2010)
13. Liu, J., Dolan, P., Pedersen, E.R.: Personalized news recommendation based on click behavior. In: Proceedings of the 15th International Conference on Intelligent User Interfaces, pp. 31–40. ACM (2010)
14. Okura, S., Tagami, Y., Ono, S., Tajima, A.: Embedding-based news recommendation for millions of users. In: Proceedings of the 23rd ACM SIGKDD, pp. 1933–1942 (2017)
15. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of the EMNLP, pp. 1532–1543 (2014)
16. Ren, H., Feng, W.: CONCERT: a concept-centric web news recommendation system. In: Wang, J., Xiong, H., Ishikawa, Y., Xu, J., Zhou, J. (eds.) WAIM 2013. LNCS, vol. 7923, pp. 796–798. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38562-9_82
17. Son, J.W., Kim, A.Y., Park, S.B.: A location-based news article recommendation with explicit localized semantic analysis. In: Proceedings of the 36th ACM SIGKDD, pp. 293–302 (2013)
18. Wang, H., Zhang, F., Xie, X., Guo, M.: DKN: deep knowledge-aware network for news recommendation. In: Proceedings of WWW, pp. 1835–1844 (2018)
19. Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X.: Neural news recommendation with attentive multi-view learning. arXiv preprint arXiv:1907.05576 (2019)
20. Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X.: NPA: neural news recommendation with personalized attention. In: Proceedings of the 25th SIGKDD, pp. 2576–2584. ACM (2019)
21. Wu, C., Wu, F., Ge, S., Qi, T., Huang, Y., Xie, X.: Neural news recommendation with multi-head self-attention. In: Proceedings of the EMNLP-IJCNLP, pp. 6390–6395 (2019)
22. Zhang, L., Liu, P., Gulla, J.A.: A deep joint network for session-based news recommendations with contextual augmentation. In: Proceedings of the 29th on Hypertext and Social Media, pp. 201–209 (2018)
23. Zhu, Q., Zhou, X., Song, Z., Tan, J., Guo, L.: Dan: deep attention neural network for news recommendation. In: Proceedings of the AAAI, vol. 33, pp. 5973–5980 (2019)
24. Zhu, Y., et al.: What to do next: modeling user behaviors by time-LSTM. In: Proceedings of the IJCAI, pp. 3602–3608 (2017)