

# Neural News Recommendation with Long- and Short-term User Representations

Mingxiao An<sup>1,\*</sup>, Fangzhao Wu<sup>2</sup>, Chuhan Wu<sup>3</sup>, Kun Zhang<sup>1</sup>, Zheng Liu<sup>2</sup>, Xing Xie<sup>2</sup>

<sup>1</sup>University of Science and Technology of China, Hefei 230026, China

<sup>2</sup>Microsoft Research Asia, Beijing 100080, China

<sup>3</sup>Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

{anmx, zhkun}@mail.ustc.edu.cn, wufangzhao@gmail.com  
wuch15@mails.tsinghua.edu.cn, {zhengliu, xingx}@microsoft.com

## Abstract

Personalized news recommendation is important to help users find their interested news and improve reading experience. A key problem in news recommendation is learning accurate user representations to capture their interests. Users usually have both long-term preferences and short-term interests. However, existing news recommendation methods usually learn single representations of users, which may be insufficient. In this paper, we propose a neural news recommendation approach which can learn both long- and short-term user representations. The core of our approach is a news encoder and a user encoder. In the news encoder, we learn representations of news from their titles and topic categories, and use attention network to select important words. In the user encoder, we propose to learn long-term user representations from the embeddings of their IDs. In addition, we propose to learn short-term user representations from their recently browsed news via GRU network. Besides, we propose two methods to combine long-term and short-term user representations. The first one is using the long-term user representation to initialize the hidden state of the GRU network in short-term user representation. The second one is concatenating both long- and short-term user representations as a unified user vector. Extensive experiments on a real-world dataset show our approach can effectively improve the performance of neural news recommendation.

## 1 Introduction

Online news platforms such as MSN News<sup>1</sup> and Google News<sup>2</sup> which aggregate news from various sources and distribute them to users have gained

\*This work was done when the first author was an intern in Microsoft Research Asia.

<sup>1</sup><https://www.msn.com/news>

<sup>2</sup><https://news.google.com/>

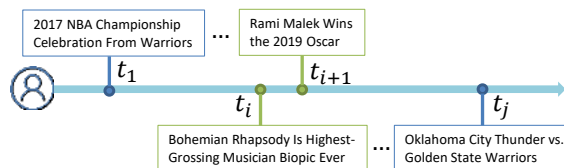


Figure 1: An illustrative example of long-term and short-term interests in news reading.

huge popularity and attracted hundreds of millions of users (Das et al., 2007; Wang et al., 2018). However, massive news are generated everyday, making it impossible for users to read through all news (Lian et al., 2018). Thus, personalized news recommendation is very important for online news platforms to help users find their interested contents and alleviate information overload (Lavie et al., 2010; Zheng et al., 2018).

Learning accurate user representations is critical for news recommendation (Okura et al., 2017). Existing news recommendation methods usually learn a single representation for each user (Okura et al., 2017; Lian et al., 2018; Wu et al., 2019). For example, Okura et al. (2017) proposed to learn representations of news using denoising autoencoder and learn representations of users from their browsed news using GRU network (Cho et al., 2014). However, it is very difficult for RNN networks such as GRU to capture the entire information of very long news browsing history. Wang et al. (2018) proposed to learn the representations of news using knowledge-aware convolutional neural network (CNN), and learn the representations of users from their browsed news based on the similarities between the candidate news and the browsed news. However, this method needs to store the entire browsing history of each user in the online news recommendation stage, which may bring huge challenge to the storage and may cause heavy latency.

Our work is motivated by the observation that the interests of online users in news are very diverse. Some user interests may last for a long time and are consistent for the same user (Li et al., 2014). For example, as shown in Fig. 1, if a user is a fan of “Golden State Warriors”, this user may tend to read many basketball news about this NBA team for several years. We call this kind of user preferences as long-term interest. In addition, many user interests may evolve with time and may be triggered by specific contexts or temporal demands. For example, in Fig. 1, the browsing of the news on movie “Bohemian Rhapsody” causes the user reading several related news such as “Rami Malek Wins the 2019 Oscar” since “Rami Malek” is an important actor in this movie, although this user may never read news about “Rami Malek” before. We call this kind of user interests as short-term interest. Thus, both long-term and short-term user interests are important for personalized news recommendation, and distinguishing long-term user interests from short-term ones may help learn more accurate user representations.

In this paper, we propose a neural news recommendation approach with both long- and short-term user representations (LSTUR). Our approach contains two major components, i.e., a news encoder and a user encoder. The news encoder is used to learn representations of news articles from their titles and topic categories. We apply attention mechanism to the news encoder to learn informative news representations by selecting important words. The user encoder consists of two modules, i.e., a long-term user representation (LTUR) module and a short-term user representation (STUR) module. In STUR, we use a GRU network to learn short-term representations of users from their recently browsing news. In LTUR, we learn the long-term representations of users from the embeddings of their IDs. In addition, we propose two methods to combine the short-term and long-term user representations. The first one is using the long-term user representations to initialize the hidden state of GRU network in the STUR model. The second one is concatenating the long-term and short-term user representations as a unified user vector. We conducted extensive experiments on a real-world dataset. The experimental results show our approach can effectively improve the performance of news recommendation and consistently outperform many baseline methods.

## 2 Related Works

Personalized news recommendation is an important task in natural language processing field and has wide applications (Zheng et al., 2018). It is critical for news recommendation methods to learn accurate news and user representations (Wang et al., 2018). Many conventional news recommendation methods rely on manual feature engineering to build news and user representations (Phelan et al., 2009; Liu et al., 2010; Li et al., 2010; Son et al., 2013; Li et al., 2014; Bansal et al., 2015; Lian et al., 2018). For example, Liu et al. (2010) proposed to use the topic categories and interests features predicted by a Bayesian model to represent news, and use the click distribution features of news categories to represent users. Li et al. (2014) used a Latent Dirichlet Allocation (LDA) (Blei et al., 2003) model to generate topic distribution features as the news representations. They represented a session by using the topic distribution of browsed news in this session, and the representations of users were built from their session representations weighted by the time. However, these methods heavily rely on manual feature engineering, which needs massive domain knowledge to craft. In addition, the contexts and orders of words in news are not incorporated, which are important for understanding the semantic meanings of news and learning representations of news and users.

In recent years, several deep learning methods were proposed for personalized news recommendation (Wang et al., 2018; Okura et al., 2017; Zheng et al., 2018). For example, Okura et al. (2017) proposed to learn representations of news from news bodies using denoising autoencoder, and learn representations of users from the representations of their browsed news using a GRU network. Wang et al. (2018) proposed to learn representations of news from their titles via a knowledge-aware CNN network, and learn representations of users from the representations of their browsed news articles weighted by their similarities with the candidate news. Wu et al. (2019) proposed to learn news and user representations with personalized word- and news-level attention networks, which exploits the embedding of user ID to generate the query vector for the attentions. However, these methods usually learn a single representation vector for each user, and cannot distinguish the long-term preferences and short-term interests of users in reading news. Thus, the user

representations learned in these methods may be insufficient for news recommendation. Different from these methods, our approach can learn both long-term and short-term user representations in a unified framework to capture the diverse interests of users for personalized neural news recommendation. Extensive experiments on the real-world dataset validate the effectiveness of our approach and the advantage over many baseline methods.

### 3 Our Approach

In this section, we present our neural news recommendation approach with long- and short-term user representations (LSTUR). Our approach contains two major components, i.e., a news encoder to learn representations of news and a user encoder to learn representations of users. Next, we introduce each component in detail.

#### 3.1 News Encoder

The *news encoder* is used to learn representations of news from their titles, topic and subtopic categories. The architecture of the *news encoder* in our approach is illustrated in Fig. 2. There are two sub-modules in the *news encoder*, i.e., a title encoder and a topic encoder.

The title encoder is used to learn news representations from titles. There are three layers in the title encoder. The first layer is word embedding, which is used to convert a news title from a word sequence into a sequence of dense semantic vectors. Denote the word sequence in a news title  $t$  as  $t = [w_1, w_2, \dots, w_N]$ , where  $N$  is the length of this title. It is transformed into  $[w_1, w_2, \dots, w_N]$  via a word embedding matrix.

The second layer in title encoder is a convolutional neural network (CNN) (LeCun et al., 2015). Local contexts are very useful for understanding the semantic meaning of news titles. For example, in the news title “Next season of super bowl games”, the local contexts of “bowl” such as “super” and “games” are very important for inferring that it belongs to a sports event name. Thus, we apply a CNN network to learn contextual word representations by capturing the local context information. Denote the contextual representation of  $w_i$  as  $c_i$ , which is computed as follows:

$$c_i = \text{ReLU}(\mathbf{C} \times \mathbf{w}_{[i-M:i+M]} + \mathbf{b}), \quad (1)$$

where  $\mathbf{w}_{[i-M:i+M]}$  is the concatenation of the embeddings of words between position  $i - M$  and

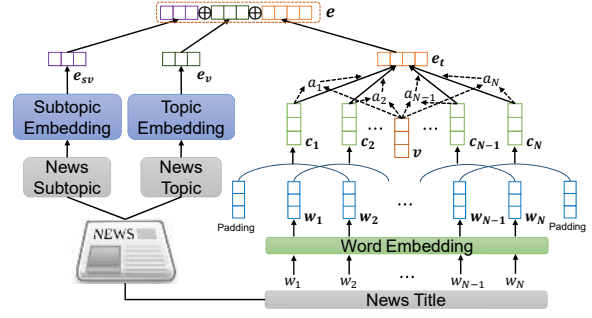


Figure 2: The framework of the news encoder.

$i + M$ .  $\mathbf{C}$  and  $\mathbf{b}$  are the parameters of the convolutional filters in CNN, and  $M$  is the window size.

The third layer is an attention network (Bahdanau et al., 2015). Different words in the same news title may have different informativeness for representing news. For instance, in the news title “The best NBA moments in 2018”, the word “NBA” is very informative for representing this news since it is an important indication of sports news, while the word “2018” is less informative. Thus, we employ a word-level attention network to select important words in news titles to learn more informative news representations. The attention weight  $\alpha_i$  of the  $i$ -th word is formulated as follows:

$$a_i = \tanh(\mathbf{v} \times \mathbf{c}_i + v_b),$$

$$\alpha_i = \frac{\exp(a_i)}{\sum_{j=1}^N \exp(a_j)}, \quad (2)$$

where  $\mathbf{v}$  and  $v_b$  are the trainable parameters. The final representation of a news title  $t$  is the summation of its contextual word representations weighted by their attention weights as follows:

$$e_t = \sum_{i=1}^N \alpha_i c_i. \quad (3)$$

The topic encoder module is used to learn news representations from its topics and subtopics. On many online news platforms such as MSN news, news articles are usually labeled with a topic category (e.g., “Sports”) and a subtopic category (e.g., “Football\_NFL”) to help target user interests. The topic and subtopic categories of news are also informative for learning representations of news and users. They can reveal the general and detailed topics of the news, and reflect the preferences of users. For example, if a user browsed many news articles with the “Sports” topic category, then we

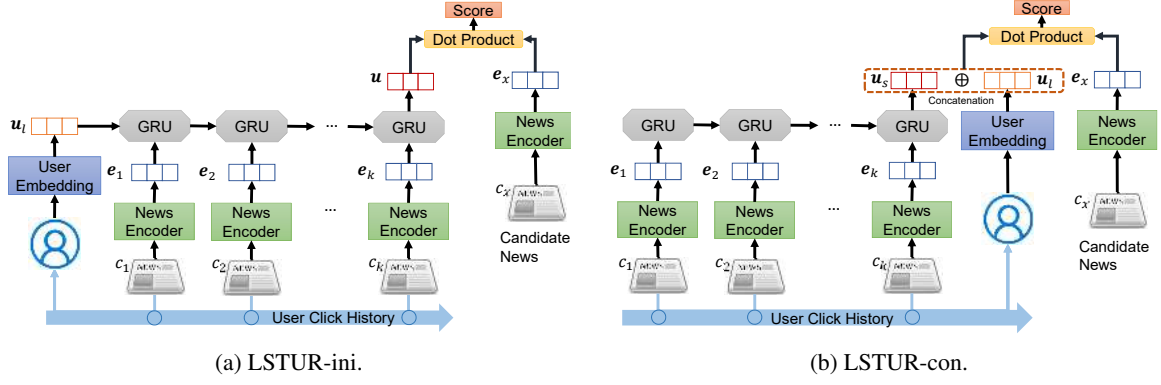


Figure 3: The two frameworks of our LSTUR approach.

can infer this user is probably interested in sports, and it may be effective to recommend candidate news in the ‘‘Sports’’ topic category to this user. To incorporate the topic and subtopic information into news representation, we propose to learn the representations of topics and subtopics from the embeddings of their IDs, as shown in Fig. 2. Denote  $e_v$  and  $e_{sv}$  as the representations of topic and subtopic. The final representation of a news article is the concatenation of the representations of its title, topic and subtopic, i.e.,  $e = [e_t, e_v, e_{sv}]$ .

### 3.2 User Encoder

The *user encoder* is used to learn representations of users from the history of their browsed news. It contains two modules, i.e., a short-term user representation model (STUR) to capture user’s temporal interests, and a long-term user representation model (LTUR) to capture user’s consistent preferences. Next, we introduce them in detail.

#### 3.2.1 Short-Term User Representation

Online users may have dynamic short-term interests in reading news articles, which may be influenced by specific contexts or temporal information demands. For example, if a user just reads a news article about ‘‘Mission: Impossible 6 – Fallout’’, and she may want to know more about the actor ‘‘Tom Cruise’’ in this movie and click news articles related to ‘‘Tom Cruise’’, although she is not his fan and may never read his news before. We propose to learn the short-term representations of users from their recent browsing history to capture their temporal interests, and use gated recurrent networks (GRU) (Cho et al., 2014) network to capture the sequential news reading patterns (Okura et al., 2017). Denote news browsing sequence from a user sorted by timestamp in ascending or-

der as  $\mathcal{C} = \{c_1, c_2, \dots, c_k\}$ , where  $k$  is the length of this sequence. We apply the news encoder to obtain the representations of these browsed articles, denoted as  $\{e_1, e_2, \dots, e_k\}$ . The short-term user representation is computed as follows:

$$\begin{aligned}
 r_t &= \sigma(\mathbf{W}_r[\mathbf{h}_{t-1}, e_t]), \\
 z_t &= \sigma(\mathbf{W}_z[\mathbf{h}_{t-1}, e_t]), \\
 \tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_{\tilde{h}}[r_t \odot \mathbf{h}_{t-1}, e_t]), \\
 \mathbf{h}_t &= z_t \odot \mathbf{h}_t + (1 - z_t) \odot \tilde{\mathbf{h}}_t,
 \end{aligned} \tag{4}$$

where  $\sigma$  is the sigmoid function,  $\odot$  is the item-wise product,  $\mathbf{W}_r$ ,  $\mathbf{W}_z$  and  $\mathbf{W}_{\tilde{h}}$  are the parameters of the GRU network. The short-term user representation is the last hidden state of the GRU network, i.e.,  $\mathbf{u}_s = \mathbf{h}_k$ .

#### 3.2.2 Long-Term User Representations

Besides the temporal interests, online users may also have long-term interests in reading news. For example, a basketball fan may tend to browse many sports news related to NBA in several years. Thus, we propose to learn long-term representations of users to capture their consistent preferences. In our approach the long-term user representations are learned from the embeddings of the user IDs, which are randomly initialized and fine-tuned during model training. Denote  $u$  as the ID of a user and  $\mathbf{W}_u$  as the look-up table for long-term user representation, the long-term user representation of this user is  $\mathbf{u}_l = \mathbf{W}_u[u]$ .

#### 3.2.3 Long- and Short-Term User Representation

In this section, we introduce two methods to combine the long-term and short-term user presentations for unified user representation, which are shown in Fig. 3.

The first method is using the long-term user representation to initialize the hidden state of the GRU network in the short-term user representation model, as shown in Fig. 3a. We denote this method as LSTUR-ini. We use the last hidden state of the GRU network as the final user representation. The second method is concatenating the long-term user representation with the short-term user representation as the final user representation, as shown in Fig. 3b. We denote this method as LSTUR-con.

### 3.3 Model Training

For online news recommendation services where user and news representations can be computed in advance, the scoring function should be as simple as possible to reduce latency. Motivated by (Okura et al., 2017), we use the simple dot production to compute the news click probability score. Denote the representation of a user  $u$  as  $\mathbf{u}$  and the representation of a candidate news article  $e_x$  as  $e_x$ , the probability score  $s(u, c_x)$  of this user clicking this news is computed as  $s(u, c_x) = \mathbf{u}^\top e_x$ .

Motivated by (Huang et al., 2013) and (Zhai et al., 2016), we propose to use the negative sampling technique for model training. For each news browsed by a user (regarded as a positive sample), we randomly sample  $K$  news articles from the same impression which are not clicked by this user as negative samples. Our model will jointly predict the click probability scores of the positive news and the  $K$  negative news. In this way, the news click prediction problem is reformulated as a pseudo  $K + 1$ -way classification task. We minimize the summation of the negative log-likelihood of all positive samples during training, which can be formulated as follows:

$$-\sum_{i=1}^P \log \frac{\exp(s(u, c_i^p))}{\exp(s(u, c_i^p)) + \sum_{k=1}^K \exp(s(u, c_{i,k}^n))}, \quad (5)$$

where  $P$  is the number of positive training samples, and  $c_{i,k}^n$  is the  $k$ -th negative sample in the same session with the  $i$ -th positive sample.

Since not all users can be incorporated in news recommendation model training (e.g., the new coming users), it is not appropriate to assume all users have long-term representations in our models in the prediction stage. In order to handle this problem, in the model training stage, we randomly mask the long-term representations of users with

a certain probability  $p$ . When we mask the long-term representations, all the dimensions are set to zero. Thus, the long-term user representation in our LSTUR approach can be reformulated as:

$$\mathbf{u}_l = M \cdot \mathbf{W}_u[u], M \sim B(1, 1-p), \quad (6)$$

where  $B$  is Bernoulli distribution, and  $M$  is a random variable that subject to  $B(1, 1-p)$ . We find in experiments that this trick for model training can improve the performance of our approach.

## 4 Experiments

### 4.1 Dataset and Experimental Settings

Since there is no off-the-shelf dataset for news recommendation, we built one by ourselves through collecting logs from MSN News<sup>3</sup> in four weeks from December 23rd, 2018 to January 19th, 2019. We used the logs in the first three weeks for model training, and those in the last week for test. We also randomly sampled 10% of logs from the training set as the validation data. For each sample, we collected the browsing history in last 7 days to learn short-term user representations. The detailed dataset statistics are summarized in Table 1.

# of users	25,000	# of users in training set	22,938
# of news	38,501	Avg. # of words per title	9.98
# of imprs	393,191	# of positive samples	492,185
NP ratio <sup>4</sup>	18.74	# of negative samples	9,224,537

Table 1: Statistics of the dataset in our experiments.

In our experiments, we used the pretrained GloVe embedding<sup>5</sup> (Pennington et al., 2014) as the initialization of word embeddings. The word embedding dimension is 200. The number of filters in CNN network is 300, and the window size of the filters in CNN network is set to 3. We applied dropout (Srivastava et al., 2014) to each layer in our approach to mitigate overfitting. The dropout rate is 0.2. The default value of long-term user representation masking probability  $p$  for model training is 0.5. We used Adam (Kingma and Ba, 2014) to optimize the model, and the learning rate was 0.01. The batch size is set to 400. The number of negative samples for each positive sample is 4. These hyper-parameters were all selected according to the results on validation set. We used

<sup>3</sup><https://www.msn.com/en-us/news>

<sup>4</sup>The ratio of the negative sample number to the positive sample number.

<sup>5</sup><http://nlp.stanford.edu/data/glove.840B.300d.zip>

impression-based ranking metrics to evaluate the performance, including area under the ROC curve (AUC), mean reciprocal rank (MRR), and normalized discounted cumulative gain (nDCG). We repeated each experiment for 10 times independently, and reported the average results with 0.95 confidence probability.

## 4.2 Performance Evaluation

We evaluate the performance of our approach by comparing it with several baseline methods, including:

- *LibFM* (Rendle, 2012), a state-of-the-art matrix factorization method which is widely used in recommendation. In our experiments, the user features are the concatenation of TF-IDF features extracted from the browsed news titles, and the normalized count features from the topics and subtopics of the browsed news. The features for news consists of TF-IDF features from its title, and one-hot vectors of its topic and subtopic. The input to *LibFM* is the concatenation of user features and features of candidate news.
- *DeepFM* (Guo et al., 2017), a widely used method that combines factorization machines and deep neural networks. We use the same features as *LibFM*.
- *Wide & Deep* (Cheng et al., 2016), another deep learning based recommendation method that combines a wide channel and a deep channel. Again, the same features with *LibFM* are used for both channels.
- *DSSM* (Huang et al., 2013), deep structured semantic model. The inputs are hashed words via character trigram, where all the browsed news titles are merged as query document.
- *CNN* (Kim, 2014), using CNN with max pooling to learn news representations from the titles of browsed news by keeping the most salient features.
- *DKN* (Wang et al., 2018), a deep news recommendation model which contains CNN and candidate-aware attention on the news browsing histories.
- *GRU* (Okura et al., 2017), learning news representations by a denoising autoencoder and user representations by a GRU network.

The results of comparing different methods are summarized in Table 2.

We have obtained observations from Table 2. First, the news recommendation methods (e.g. *CNN*, *DKN* and *LSTUR*) which use neural networks to learn news and user representations can significantly outperform the methods using manual feature engineering (e.g. *LibFM*, *DeepFM*, *Wide & Deep*, and *DSSM*). This is probably because handcrafted features are usually not optimal, and neural networks can capture both global and local semantic contexts in news, which are useful for learning more accurate news and user representations for news recommendation.

Second, our *LSTUR* approach outperforms all baseline methods compared here, including deep learning models such as *CNN*, *GRU* and *DKN*. Our *LSTUR* approach can capture both the long-term preferences and short-term interests to capture the complex and diverse user interests in news reading, while the baseline methods only learn a single representation for each user, which is insufficient. In addition, our *LSTUR* approach uses attention mechanism in the news encoder to select important words, which can help learn more informative news representations.

Third, our proposed two methods to learn long- and short-term user representations, i.e., *LSTUR*-ini and *LSTUR*-con, can achieve comparable performance and both outperform baseline methods, which validate the effectiveness of these methods. In addition, the performance of *LSTUR*-con is more stable than *LSTUR*-ini, which indicates that using the concatenation of both short-term and long-term user representations is capable of retaining all the information. We also conducted experiments to explore the performance of combining both *LSTUR*-con and *LSTUR*-ini in the same model, but the performance improvement is very limited, implying that each of them can fully capture the long- and short-term user interests for news recommendation.

## 4.3 Effectiveness of Long- and Short-Term User Representation

In this section, we conducted several experiments to explore the effectiveness of our approach in learning both long-term and short-term user representations. We compare the performance of our *LSTUR* methods with the long-term user representation model *LTUR* and the short-term user rep-

Methods	AUC	MRR	nDCG@5	nDCG@10
LibFM	56.52 $\pm$ 1.31	25.53 $\pm$ 0.81	26.66 $\pm$ 1.04	34.72 $\pm$ 0.95
DeepFM	58.13 $\pm$ 1.69	27.01 $\pm$ 0.20	28.37 $\pm$ 0.57	36.78 $\pm$ 0.62
Wide & Deep	58.07 $\pm$ 0.55	27.07 $\pm$ 0.37	28.51 $\pm$ 0.45	36.93 $\pm$ 0.43
DSSM	58.43 $\pm$ 0.58	27.25 $\pm$ 0.49	28.31 $\pm$ 0.60	36.91 $\pm$ 0.54
CNN	61.13 $\pm$ 0.77	29.44 $\pm$ 0.73	31.44 $\pm$ 0.87	39.51 $\pm$ 0.74
DKN	61.25 $\pm$ 0.78	29.47 $\pm$ 0.64	31.54 $\pm$ 0.79	39.59 $\pm$ 0.67
GRU	62.69 $\pm$ 0.16	30.24 $\pm$ 0.13	32.56 $\pm$ 0.17	40.55 $\pm$ 0.13
LSTUR-con	63.47 $\pm$ 0.10	30.94 $\pm$ 0.14	33.43 $\pm$ 0.13	41.34 $\pm$ 0.13
LSTUR-ini	<b>63.56 <math>\pm</math> 0.42</b>	<b>30.98 <math>\pm</math> 0.32</b>	<b>33.45 <math>\pm</math> 0.39</b>	<b>41.37 <math>\pm</math> 0.36</b>

Table 2: The performance of different methods on news recommendation.

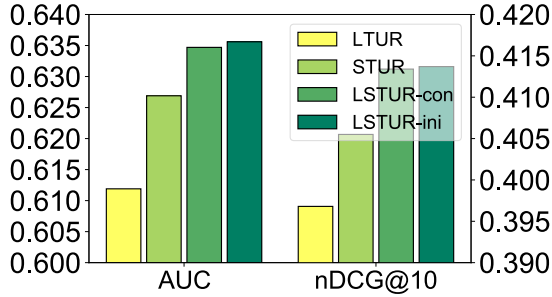


Figure 4: The effectiveness of incorporating long-term user representations (LTUR) and short-term user representations (STUR).

representation model STUR. The results are summarized in Fig. 4.

From the results we find both LTUR and STUR are useful for news recommendation, and the STUR model can outperform the LTUR model. According to the statistics in Table 1, the long-term representations of many users in test data are unavailable, which leads to relative weak performance of LTUR on these users. In addition, combining STUR and LTUR using our two long- and short-term user representation methods, i.e., LSTUR-ini and LSTUR-con, can effectively improve the performance. This result validates that incorporating both long-term and short-term user representations is useful to capture the diverse user interests more accurately and is beneficial for news recommendation.

#### 4.4 Effectiveness of News Encoders in STUR

In our STUR model, GRU is used to learn short-term user representations from the recent browsing news. We explore the effectiveness of GRU in encoding news by replacing it with several other encoders, including: 1) Average: using the average of all the news representations in recent browsing

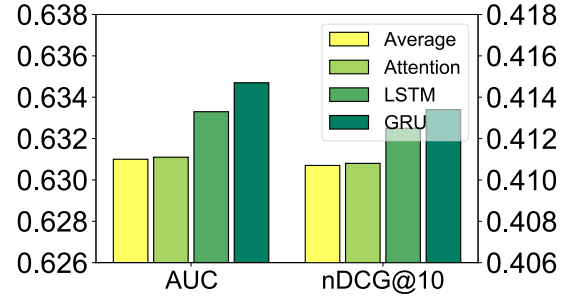


Figure 5: The comparisons of different methods in learning short-term user representations from recently browsed news articles.

history; 2) Attention: the summation of news representations weighted by their attention weights; 3) LSTM (Hochreiter and Schmidhuber, 1997), replacing GRU with LSTM. The results are summarized in Fig. 5.

According to Fig. 5, the sequence-based encoders (e.g., GRU, LSTM) outperform the Average and Attention based encoders. This is probably because the sequence-based encoders can capture the sequential new reading patterns to learn short-term representations of users, which is difficult for Average and Attention based encoders. In addition, GRU achieves better performance than LSTM. This may be because GRU contains fewer parameters and has lower risk of overfitting. Thus, we select GRU as the news encoder in STUR.

#### 4.5 Effectiveness of News Title Encoders

In this section, we conduct experiments to compare different news title encoders. In our approach, the news encoder is a combination of CNN network and an attention network (denoted as CNN+Att). We compare it with several variants, i.e., CNN, LSTM, and LSTM with attention (LSTM+Att), to validate the effectiveness of our

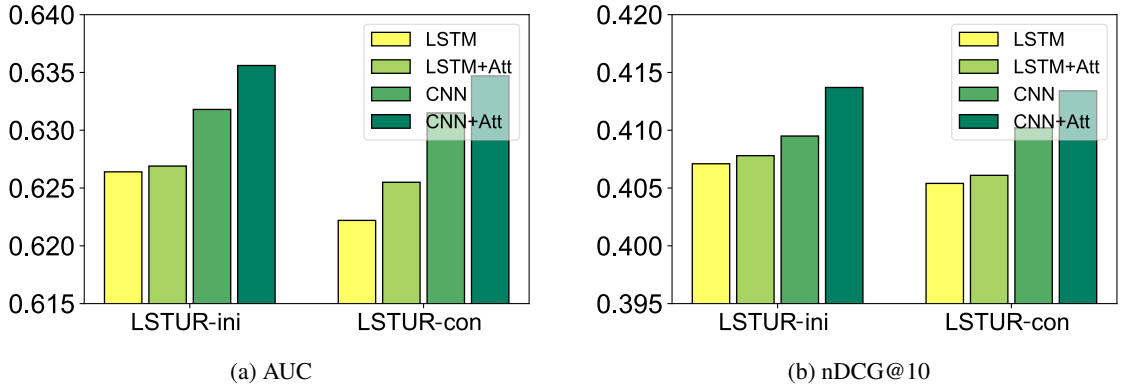


Figure 6: The comparisons of different methods in learning news title representations and the effectiveness of attention mechanism in selecting important words.

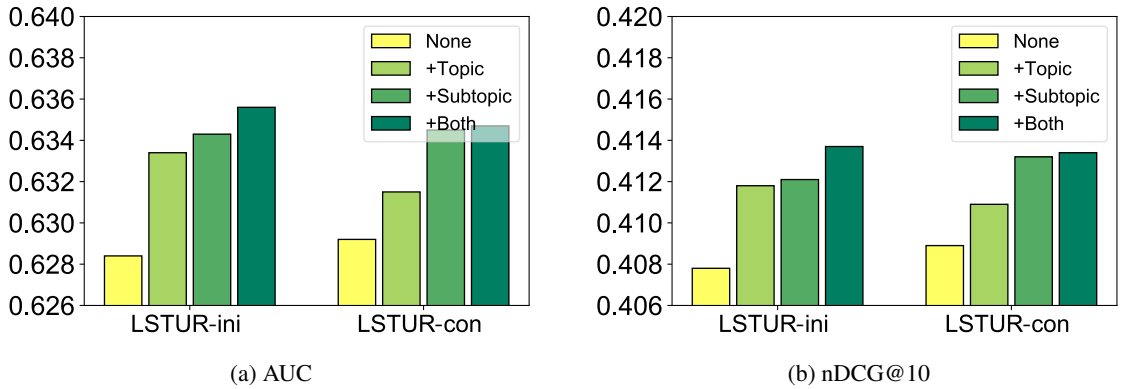


Figure 7: The effectiveness of incorporating news topic and subtopic information for news recommendation.

approach. The results are summarized in Fig. 6.

According to Fig. 6, using attention mechanism in both encoders based on CNN and LSTM can achieve better performance. This is probably because the attention network can select important words, which can learn more informative news representations. In addition, encoders using CNN outperform those using LSTM. This may be because local contexts in news titles are more important for learning news representations.

#### 4.5.1 Effectiveness of News Topic

In this section, we conduct experiments to validate the effectiveness of incorporating topic and subtopic of news in the news encoder. We compare the performance of our approach with its variants without topic and/or subtopics. The results are shown in Fig. 7.

According to Fig. 7, incorporating either topics or subtopics can effectively improve the performance of our approach. In addition, the news encoder with subtopics outperforms the news encoder with topics. This is probably because subtopics can provide more fine-grained topic in-

formation which is more helpful for news recommendation. Thus, the model with subtopics can achieve better news recommendation performance. Moreover, combining topics and subtopics can further improve the performance of our approach. These results validate the effectiveness of our approach in exploiting topic information for news recommendation.

#### 4.5.2 Influence of Masking Probability

In this section, we explore the influence of the probability  $p$  in Eq. (6) for randomly masking long-term user representation in model training. We vary the value of  $p$  from 0.0 to 0.9 with a step of 0.1 for both LSTUR-ini and LSTUR-con. The results are summarized in Fig. 8.

According to Fig. 8, the results of LSTUR-ini and LSTUR-con have similar patterns. The performance of both methods improves when  $p$  increases from 0. When  $p$  is too small, the model will tend to overfit on the LTUR, since LTUR has many parameters. Thus, the performance is not optimal. However, when  $p$  is too large, the performance of both methods starts to decline. This may be be-



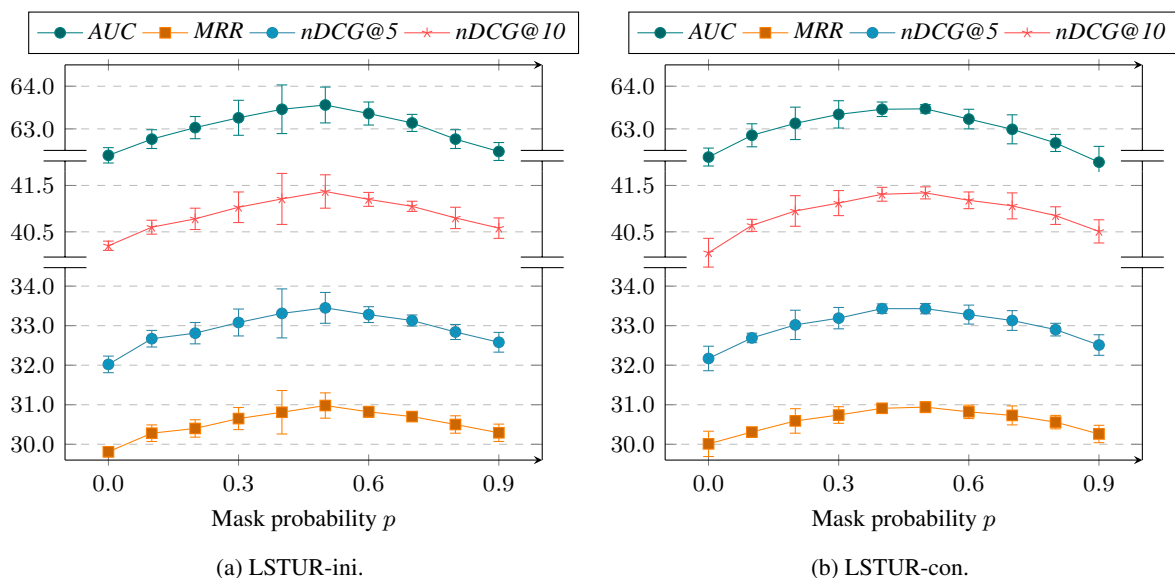


Figure 8: The influence of mask probability  $p$  on the performance of our approach.

2019 CES Highlights : Innovations in Enviro-Sensing for Robocars  
 California dries off after storm batter state for days  
 15 Recipes Inspired By Vintage Movies  
 Texas State Rep . Dennis Bonnen Elected As House Speaker  
 Should You Buy American Express Stock After Earnings ?  
 How Meghan Markle Has Changed Prince Harry Considerably

Figure 9: Visualization of the word-level attentions.

cause the useful information in LTUR cannot be effectively incorporated. Thus, the performance is also not optimal. A moderate choice on  $p$  (e.g., 0.5) is most appropriate for both LSTUR-ini and LSTUR-con methods, which can properly balance the learning of LTUR and STUR.

## 5 Visualization of Attention Weights

In this section, we visually explore the effectiveness of the word-level attention network in the news encoder. The attention weights in several example news titles are shown in Fig. 9. From the results, we find our approach can effectively recognize important words to learn more informative news representations. For example, the words “Enviro-Sensing” and “Robocars” in the first news title are assigned high attention weights because these words are indications of news on technologies, while the words “2019” and “for” are assigned low attention weights by our approach since they are less informative. These results validate the effectiveness of the attention network in the news encoder.

## 6 Conclusion

In this paper, we propose a neural news recommendation approach which can learn both long- and short-term user representations. The core of our model is a news encoder and a user encoder. In the news encoder, we learn representations of news from their titles and topic categories, and use an attention network to highlight important words for informative representation learning. In the user encoder, we propose to learn long-term representations of users from the embeddings of their IDs. In addition, we learn short-term representations of users from their recently browsed news via a GRU network. Besides, we propose two methods to fuse long- and short-term user representations, i.e., using long-term user representation to initialize the hidden state of the GRU network in short-term user representation, or concatenating both long- and short-term user representations as a unified user vector. Extensive experiments on a real-world dataset collected from MSN news show our approach can effectively improve the performance of news recommendation.

## Acknowledgement

The authors would like to thank Microsoft News for providing technical support and data in the experiments, and Jiun-Hung Chen (Microsoft News) and Ying Qiao (Microsoft News) for their support and discussions. We also want to thank Jianqiang Huang for his help in the experiments.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. [Neural machine translation by jointly learning to align and translate](#). In *ICLR*.
- Trapit Bansal, Mrinal Das, and Chiranjib Bhattacharyya. 2015. [Content driven user profiling for comment-worthy recommendations of news and blog articles](#). In *RecSys*, pages 195–202.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. [Latent dirichlet allocation](#). *Journal of Machine Learning Research*, 3(Jan):993–1022.
- Heng-Tze Cheng, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhya, Glen Anderson, Greg Corrado, and Wei Chai. 2016. [Wide & deep learning for recommender systems](#). In *DLRS*, pages 7–10.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. [Learning phrase representations using RNN encoder–decoder for statistical machine translation](#). In *EMNLP*, pages 1724–1734.
- Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. [Google news personalization: scalable online collaborative filtering](#). In *WWW*, pages 271–280.
- Huifeng Guo, Ruiming TANG, Yunming Ye, Zhengguo Li, and Xiuqiang He. 2017. [DeepFM: A factorization-machine based neural network for CTR prediction](#). In *IJCAI*, pages 1725–1731.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#). *Neural Computation*, 9(8):1735–1780.
- Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. [Learning deep structured semantic models for web search using clickthrough data](#). In *CIKM*, pages 2333–2338.
- Yoon Kim. 2014. [Convolutional neural networks for sentence classification](#). In *EMNLP*, pages 1746–1751.
- Diederik P Kingma and Jimmy Ba. 2014. [Adam: A method for stochastic optimization](#). *arXiv preprint arXiv:1412.6980*.
- Talia Lavie, Michal Sela, Ilit Oppenheim, Ohad Inbar, and Joachim Meyer. 2010. [User attitudes towards news content personalization](#). *International Journal of Human-Computer Studies*, 68(8):483–495.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. [Deep learning](#). *Nature*, 521(7553):436–444.
- Lei Li, Li Zheng, Fan Yang, and Tao Li. 2014. [Modeling and broadening temporal user interest in personalized news recommendation](#). *Expert Systems with Applications*, 41(7):3168–3177.
- Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. [A contextual-bandit approach to personalized news article recommendation](#). In *WWW*, pages 661–670.
- Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2018. [Towards better representation learning for personalized news recommendation: a multi-channel deep fusion approach](#). In *IJ-CAI*, pages 3805–3811.
- Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. [Personalized news recommendation based on click behavior](#). In *IUI*, pages 31–40.
- Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. [Embedding-based news recommendation for millions of users](#). In *KDD*, pages 1933–1942.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [Glove: Global vectors for word representation](#). In *EMNLP*, pages 1532–1543.
- Owen Phelan, Kevin McCarthy, and Barry Smyth. 2009. [Using twitter to recommend real-time topical news](#). In *RecSys*, pages 385–388.
- Steffen Rendle. 2012. [Factorization machines with libFM](#). *ACM Transactions on Intelligent Systems and Technology*, 3(3):1–22.
- Jeong-Woo Son, A-Yeong Kim, and Seong-Bae Park. 2013. [A location-based news article recommendation with explicit localized semantic analysis](#). In *SI-GIR*, pages 293–302.
- Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. [Dropout: a simple way to prevent neural networks from overfitting](#). *Journal of Machine Learning Research*, 15(1):1929–1958.
- Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. [DKN: Deep knowledge-aware network for news recommendation](#). In *WWW*, pages 1835–1844.
- Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. [NPA: Neural news recommendation with personalized attention](#). In *KDD*.
- Shuangfei Zhai, Keng hao Chang, Ruofei Zhang, and Zhongfei Mark Zhang. 2016. [Deepintert: Learning attentions for online advertising with recurrent neural networks](#). In *KDD*, pages 1295–1304.
- Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. [DRN: A deep reinforcement learning framework for news recommendation](#). In *WWW*, pages 167–176.