NPA: Neural News Recommendation with Personalized Attention

Chuhan Wu
Tsinghua University
Beijing, China
wuch15@mails.tsinghua.edu.cn

Fangzhao Wu
Microsoft Research Asia
Beijing, China
wufangzhao@gmail.com

Mingxiao An
USTC
Hefei, Anhui
anmx@mail.ustc.edu.cn

Jianqiang Huang
Peking University
Beijing, China
1701210864@pku.edu.cn

Yongfeng Huang
Tsinghua University
Beijing, China
yfhuang@tsinghua.edu.cn

Xing Xie
Microsoft Research Asia
Beijing, China
xingx@microsoft.com

ABSTRACT

News recommendation is very important to help users find interested news and alleviate information overload. Different users usually have different interests and the same user may have various interests. Thus, different users may click the same news article with attention on different aspects. In this paper, we propose a neural news recommendation model with personalized attention (NPA). The core of our approach is a news representation model and a user representation model. In the news representation model we use a CNN network to learn hidden representations of news articles based on their titles. In the user representation model we learn the representations of users based on the representations of their clicked news articles. Since different words and different news articles may have different informativeness for representing news and users, we propose to apply both word- and news-level attention mechanism to help our model attend to important words and news articles. In addition, the same news article and the same word may have different informativeness for different users. Thus, we propose a personalized attention network which exploits the embedding of user ID to generate the query vector for the word- and news-level attentions. Extensive experiments are conducted on a real-world news recommendation dataset collected from MSN news, and the results validate the effectiveness of our approach on news recommendation.

KEYWORDS

News Recommendation, Neural Network, Personalized Attention

ACM Reference Format:
Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: Neural News Recommendation with Personalized Attention. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330665

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330665

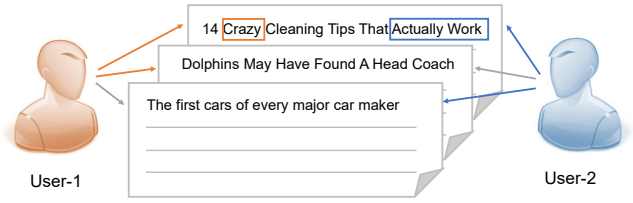


Figure 1: An illustrative example of two users and their clicked news articles. The colored arrows and boxes represent their interested news and words respectively.

1 INTRODUCTION

Online news platforms such as MSN News and Google News have attracted a huge number of users to read digital news [7, 18]. However, massive news articles are emerged everyday, and it is impractical for users to seek for interested news from a huge volume of online news articles [26, 34]. Therefore, it is an important task for online news platforms to target user interests and make personalized news recommendation [1, 8, 14], which can help users to find their interested news articles and alleviate information overload [32, 35].

There are two common observations in the news recommendation scenario. First, not all news clicked by users can reflect the preferences of users. For example, as illustrated in Figure 1, user-1 clicked all the three news, but he/she was only interested in the first and the second news. In addition, the same news should also have different informativeness for modeling different users. For example, if user-1 is very interested in sports news but user-2 rarely reads, the news "Dolphins May Have Found A Head Coach" is very informative for characterizing user-1, but less informative for user-2. Second, different words in news titles usually have different informativeness for learning news representations. For example, the word "Crazy" in the first news title is informative, while the word "That" is uninformative. Moreover, the same words within a news title may also have different informativeness for revealing preferences of different users. For example, user-1 may be attracted by the word "Crazy", and user-2 may pay more attention to the words "Actually Work". Therefore, modeling the different informativeness of words and news for different users may be useful for learning better representations of users for accurate news recommendation.

Existing news recommendation methods are usually based on collaborative filtering (CF) techniques and news content[20–22, 24]. For example, Liu et al. [21] proposed a CF-based approach for news recommendation based on user interests. They use a Bayesian model to extract the interest features of users based on the click distributions on news articles in different categories. Okura et al. [24] proposed to first learn the distributed representations of news articles based on similarity and then use recurrent neural networks to learn user representations from browsing histories for click prediction. Lian et al. [20] proposed a deep fusion model (DMF) to learn representations of users and news using combinations of fully connected networks with different depth. They also used attention mechanism to select important user features. However, all these existing methods cannot model the different informativeness of news and their words for different users, which may be useful for improving the quality of personalized news recommendation.

In this paper, we propose a neural approach with personalized attention (NPA) for news recommendation. The core of our approach is a news representation model and a user representation model. In the news representation model we use a CNN network to learn the contextual representations of news titles, and in the user representation model we learn representations of users from their clicked news. Since different words and news articles usually have different informativeness for learning representations of news and users, we propose to apply attention mechanism at both word- and news-level to select and highlight informative words and news. In addition, since the informativeness of the same words and news may be different for different users, we propose a personalized attention network by using the embedding of user ID as the query vector of the word- and news-level attention networks to differentially attend to important words and news according to user preferences. Extensive experiments on a real-world dataset collected from MSN news validate the effectiveness of our approach on news recommendation.

2 RELATED WORK

2.1 News Recommendation

News recommendation is an important task in the data mining field, and have been widely explored over years. Traditional news recommendation methods usually based on news relatedness [23], semantic similarities [3] and human editors' demonstration [33]. However, the preferences of users cannot be effectively modeled. Therefore, most news recommendation methods are based on CF techniques. The earliest study on CF methods for news recommendation is the Grouplens project [17], which applied CF methods to aggregate news from Usenet. However, pure CF methods usually suffer from the sparsity and the cold-start problems, which are especially significant in news recommendation scenarios [19]. Thus, content-based techniques are usually complementary methods to CF [2, 20, 21, 24, 27, 29, 32, 39**?**]. For example, Liu et al. [21] proposed to incorporate user interests for news recommendation. They use a Bayesian model to predict the interests of users based on the distributions of their clicked news articles in different categories. Okura et al. [24] proposed to learn news embeddings based on the similarities between news articles in the same and different categories. They use recurrent neural networks to learn user representations from the browsing histories through time to predict the score of news. Lian et al. [20] proposed a deep fusion model (DMF) to learn representations of users from various features extracted from their news reading, general web browsing and web searching histories. They used an inception network with attention mechanism to select important user features and combine the features learned by networks with different depths. Wang et al. [32] proposed to use the embeddings of the entities extracted from a knowledge graph as a separate channel of the CNN input. However, these existing methods cannot simultaneously model the informativeness of words and news. Different from all these methods, we propose to use personalized attention mechanism at both word- and new-level to dynamically recognize different informative words and news according to the preference of different users. Experimental results validate the effectiveness of our approach.

2.2 Neural Recommender Systems

In recent years, deep learning techniques have been widely used in recommender systems [31]. For example, Xue et al. [37] proposed to use multi-layer neural networks to learn the latent factors of users and items in matrix factorization. However, the content of users and items cannot be exploited, which is usually important for recommender systems. Different from using neural networks within traditional matrix factorization frameworks, many methods apply neural networks to learn representations of users and items from raw features [5, 6, 9, 11–13]. For example, Huang et al. [13] proposed a deep structured semantic model (DSSM) for click-through rate (CTR) prediction. They first hashed the very sparse bag-of-words vectors into low-dimensional feature vectors based on character n-grams, then used multi-layer neural networks to learn the representations of query and documents, and jointly predicted the click score of multiple documents. Cheng et al. [5] proposed a Wide & Deep approach to combine a wide channel using a linear transformer with a deep channel using multi-layer neural networks. Guo et al. [11] proposed a DeepFM approach which combines factorization machines with deep neural networks. The two components share the same input features and the final predicted score is calculated from the combination of the output from both components. However, these methods usually rely on handcrafted features, and the dimension of feature vectors is usually huge. In addition, they cannot effectively recognize the important contexts when learning news and user representations. Different from the aforementioned methods, our approach can dynamically select important words and news for recommendation based on user preferences, which may be useful for learning more informative user representations for personalized news recommendation.

3 OUR APPROACH

In this section, we introduce our *NPA* approach with personalized attention for news recommendation. There are three major modules in our model. The first one is a *news encoder*, which aims to learn the representations of news. The second one is a *user encoder*, which aims to learn user representations based on the representations of his/her clicked news. The third one is a *click predictor*, which is used to predict the click score of a series of candidate news. In

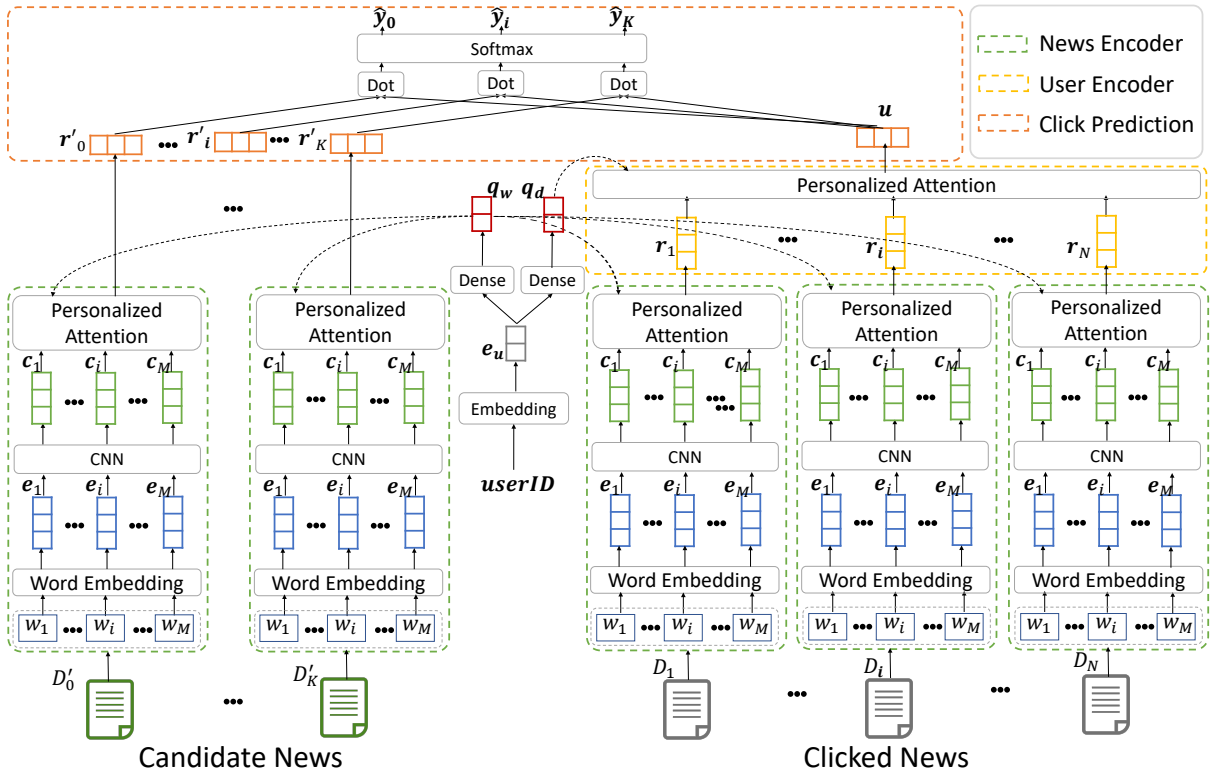


Figure 2: The framework of our NPA approach for news recommendation.

the *news encoder* and *user encoder* module, we apply personalized attention networks at both word- and new-level to differentially select informative words and news according to user preferences. The architecture of our approach is shown in Figure 2. We will introduce the details of our approach in the following sections.

3.1 News Encoder

Since users' click decisions on news platforms are usually made based on the titles of news articles, in our approach the *news encoder* module aims to learn news representations from news titles. As shown in Figure 2, there are three sub-modules in the *news encoder* module.

The first one is word embedding. It is used to convert a sequence of words within news title into a sequence of low-dimensional dense vectors. We denote the word sequence of the news D_i as $D_i = [w_1, w_2, ..., w_M]$, where M is the number of words in the news title. The word sequence D_i is transformed into a sequence of vector $\mathbf{E}^w = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_M]$ using a word embedding matrix $\mathbf{W}_e \in \mathcal{R}^{V \times D}$, where V denotes the vocabulary size and D denotes the dimension of word embedding vectors.

The second one is a convolutional neural network (CNN) [15]. CNN is an effective neural architecture for capturing local information [36]. Usually, local contexts within news are important for news recommendation. For example, in the news title "best Fiesta bowl moments", the local combination of the words "Fiesta" and "bowl" is very important to infer the topic of this news. Thus, we apply a CNN network to the word sequences to learn contextual representations of words within news titles by capturing their local contexts. Denote the representation of the i-th word as $\mathbf{c}_i$, which is calculated as:

$$\mathbf{c}_i = \text{ReLU}(\mathbf{F}_w \times \mathbf{e}_{(i-k):(i+k)} + \mathbf{b}_w), \quad (1)$$

where $\mathbf{e}_{(i-k):(i+k)}$ denotes the concatenation of the word embeddings from the position $(i-k)$ to $(i+k)$. $\mathbf{F}_w \in \mathcal{R}^{N_f \times (2k+1)D}$ and $\mathbf{b}_w \in \mathcal{R}^{N_f}$ denote the parameters of the CNN filters, where N_f is the number of CNN filters and $2k+1$ is their window size. ReLU [10] is used as the non-linear function for activation. The output of the CNN layer is the sequence of contextual word representation vectors, denoted as $[\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_M]$.

The third one is a word-level personalized attention network. Different words in a news title usually have different informativeness for characterizing the topics of news. For example, in the news entitled with "NBA games in this season" the word "NBA" is very informative for learning news representations, since it conveys important clues about the news topic, while the word "this" is less informative for recommendation. In addition, the same word may also have different informativeness for the recommendation of different users. For example, in the news title "Genesis G70 is the 2019 MotorTrend Car of the Year", the words "Genesis" and "MotorTrend" are informative for the recommendation of users who are interested in cars, but may be less informative for users who are not interested. Thus, recognizing important words for different users is useful for news recommendation. However, in vanilla non-personalized attention networks [20], the attention weights

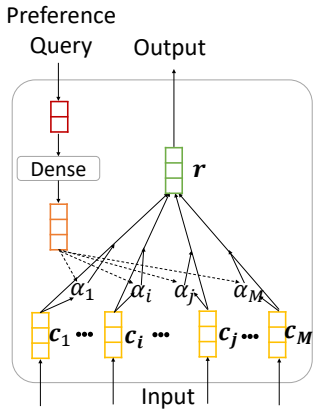


Figure 3: The architecture of the personalized attention module in our *NPA* approach.

are only calculated based on the input representation sequence via a fixed attention query vector, and the user preferences are not incorporated. To model the informativeness of each word for the recommendation of different users, we propose to use a personalized attention network to recognize and highlight important words within news titles according to user preferences. The architecture of our personalized attention module is shown in Figure 3. In order to obtain the representation of user preferences, we first embed the ID of users into a representation vector $\mathbf{e}_u \in \mathcal{R}^{D_e}$, where D_e denotes the size of user embedding. Then we use a dense layer to learn the word-level user preference query $\mathbf{q}_w$ as:

$$\mathbf{q}_w = \text{ReLU}(\mathbf{V}_w \times \mathbf{e}_u + \mathbf{v}_w), \tag{2}$$

where $\mathbf{V}_w \in \mathcal{R}^{D_e \times D_q}$ and $\mathbf{v}_w \in \mathcal{R}^{D_q}$ are parameters, D_q is the preference query size.

In this module, the attention weight of each word is calculated based on the interactions between the preference query and word representations. We denote the attention weight of the i-th word as α_i, which is formulated as:

$$a_i = \mathbf{c}_i^T \tanh(\mathbf{W}_p \times \mathbf{q}_w + \mathbf{b}_p), \tag{3}$$

$$\alpha_i = \frac{\exp(a_i)}{\sum_{j=1}^{M} \exp(a_j)}, \tag{4}$$

where $\mathbf{W}_p \in \mathcal{R}^{D_q \times N_f}$ and $b_p \in \mathcal{R}^{N_f}$ are projection parameters. The final contextual representation $\mathbf{r}_i$ of the i-th news title is the summation of the contextual representations of words weighted by their attention weights:

$$\mathbf{r}_i = \sum_{j=1}^{M} \alpha_j \mathbf{c}_j. \tag{5}$$

We apply the *news encoder* to all users' clicked news and candidate news. The representations of clicked news of a user and candidate news are respectively denoted as $[\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N]$ and $[\mathbf{r}'_0, \mathbf{r}'_1, ..., \mathbf{r}'_K]$, where N is the number of clicked news and $K+1$ is the number of candidate news.

3.2 User Encoder

The *user encoder* module in our approach aims to learn the representations of users from the representations of their clicked news, as shown in Figure 2. In this module, a news-level personalized attention module is used to build informative user representations. Usually the news clicked by the same user have different informativeness for learning user representations. For example, the news "10 tips for cooking" is very informative for modeling user preferences, but the news "It will be Rainy next week" is less informative. In addition, the same news also has different informativeness for modeling different users. For example, the news "100 Greatest Golf Courses" is informative for characterizing users who are interested in golf, but is noisy for modeling users who are actually not interested in. To model the different informativeness of the same news for different users, we also apply personalized attention mechanism to the representations of news clicked by the same user. Similar with the word-level attention network, we first transform the user embedding vector into a news preference query $\mathbf{q}_d$, which is formulated as:

$$\mathbf{q}_d = \text{ReLU}(\mathbf{V}_d \times \mathbf{e}_u + \mathbf{v}_d), \tag{6}$$

where $\mathbf{V}_d \in \mathcal{R}^{D_e \times D_d}$ and $\mathbf{v}_d \in \mathcal{R}^{D_d}$ are parameters, D_d is the dimension of the news preference query.

We denote the attention weight of the i-th news as α'_i, which is calculated by evaluating the importance of the interactions between the news preference query and news representations as follows:

$$a'_i = \mathbf{r}_i^T \tanh(\mathbf{W}_d \times \mathbf{q}_d + b_d), \tag{7}$$

$$\alpha'_i = \frac{\exp(a'_i)}{\sum_{j=1}^{N} \exp(a'_j)}, \tag{8}$$

where $\mathbf{W}_d \in \mathcal{R}^{D_d \times N_f}$ and $b_d \in \mathcal{R}^{N_f}$ are projection parameters. The final user representation $\mathbf{u}$ is the summation of the news contextual representations weighted by their attention weights:

$$\mathbf{u} = \sum_{j=1}^{N} \alpha'_j \mathbf{r}_j. \tag{9}$$

3.3 Click Predictor

The *click predictor* module is used to predict the click score of a user on each candidate news. A common observation in news recommendation is that most users usually only click a few news displayed in an impression. Thus, the number of positive and negative news samples is highly imbalanced. In many neural news recommendation methods [20, 32], the model only predicts the click score for a single piece of news (the sigmoid activation function is usually used in these methods). In these methods, positive and negative news samples are manually balanced by randomly sampling, and the rich information provided by negative samples is lost. In addition, since the total number of news samples is usually huge, the computational cost of these methods is usually heavy during model training. Thus, these methods are sub-optimal for simulating real-world news recommendation scenarios. Motivated by [13] and [38], we propose to apply negative sampling techniques by jointly predicting the click score for $K+1$ news during model training to solve the two problems above. The $K+1$ news consist of one positive sample of a user, and K randomly selected negative samples of a

user. The score $\hat{y}_i$ of the candidate news D'_i is calculated by the inner product of the news and user representation vector first, and then normalized by the softmax function, which is formulated as:

$$\hat{y}'_i = \mathbf{r}_i^{'T}\mathbf{u}, \tag{10}$$

$$\hat{y}_i = \frac{\exp(\hat{y}'_i)}{\sum_{j=0}^{K}\exp(\hat{y}'_j)} \tag{11}$$

For model training, we formulate the click prediction problem as a pseudo $K+1$ way classification task, i.e., the clicked news is the positive class and all the rest news are negative classes. We apply maximum likelihood method to minimize the log-likelihood on the positive class:

$$\mathcal{L} = -\sum_{y_j \in \mathcal{S}} \log(\hat{y}_j), \tag{12}$$

where y_j is the gold label, $\mathcal{S}$ is the set of the positive training samples. By optimizing the loss function $\mathcal{L}$ via gradient descend, all parameters can be tuned in our model. Compared with existing news recommendation methods, our approach can effectively exploit the useful information in negative samples, and further reduce the computational cost for model training (nearly divided by K). Thus, our model can be trained more easily on a large collection of news click logs.

4 EXPERIMENTS

4.1 Datasets and Experimental Settings

Our experiments were conducted on a real-world dataset, which was constructed by randomly sampling user logs from MSN News[1] in one month, i.e., from December 13rd, 2018 to January 12nd, 2019. The detailed statistics of the dataset is shown in Table 1[2]. We use the logs in the last week as the test set, and the rest logs are used for training. In addition, we randomly sampled 10% of samples for validation.

In our experiments, the dimension D of word embedding was set to 300. we used the pre-trained Glove embedding[3] [25], which is trained on a corpus with 840 billion tokens, to initialize the embedding matrix. The number of CNN filters N_f was set to 400, and the window size was 3. The dimension of user embedding D_e was set to 50. The sizes of word and news preferences queries (D_q and D_d) were both set to 200. The negative sampling ratio K was set to 4. We randomly sampled at most 50 browsed news articles to learn user representations. We applied dropout strategy [30] to each layer in our approach to mitigate overfitting. The dropout rate was set to 0.2. Adam [16] was used as the optimization algorithm for gradient descend. The batch size was set to 100. Due to the limitation of GPU memory, the maximum number of clicked news for learning user representations was set to 50 in neural network based methods, and the maximum length of news title was set to 30. These hyperparameters were selected according to the validation set. The metrics in our experiments include the average AUC, MRR, nDCG@5 and nDCG@10 scores over all impressions. We independently repeated each experiment for 10 times and reported the average performance.

[1] https://www.msn.com/en-us/news
[2] All words are lower-cased.
[3] http://nlp.stanford.edu/data/glove.840B.300d.zip

Table 1: Statistics of our dataset. *Denote the ratio of the negative sample number to positive sample number.

# users	10,000	**avg. # words per title**	11.29
# news	42,255	**# positive samples**	489,644
# impressions	445,230	**# negative samples**	6,651,940
# samples	7,141,584	**NP ratio***	13.59

4.2 Performance Evaluation

First, we will evaluate the performance of our approach by comparing it with several baseline methods. The methods to be compared include:

- *LibFM* [28], which is a state-of-the-art feature-based matrix factorization and it is a widely used method for recommendation. In our experiments, we extract the TF-IDF features from users' clicked news and candidate news, and concatenate both types of features as the input for LibFM.
- *CNN* [15], applying CNN to the word sequences in news titles and use max pooling technique to keep the most salient features, which is widely used in content-based recommendation [4, 40].
- DSSM [13], a deep structured semantic model with word hashing via character trigram and multiple dense layers. In our experiments, we concatenate all user's clicked news as a long document as the query, and the candidate news are documents. The negative sampling ratio was set to 4.
- *Wide & Deep* [5], using the combination of a wide channel using a linear transformer and a deep channel with multiple dense layers. The features we used are the same with *LibFM* for both channels.
- *DeepFM* [11], which is also a widely used neural recommendation method, using a combination with factorization machines and deep neural networks. We use the same TF-IDF features to feed for both components.
- *DFM* [20], a deep fusion model by using combinations of dense layers with different depths. We use both TF-IDF features and word embeddings as the input for DFM.
- *DKN* [32], a deep news recommendation method with Kim CNN and news-level attention network. They also incorporated entity embeddings derived from knowledge graphs.
- *NPA*, our neural news recommendation approach with personalized attention.

The experimental results on news recommendation are summarized in Table 2. According to Table 2, We have several observations.

First, the methods based on neural networks (e.g., *CNN*, *DSSM* and *NPA*) outperform traditional matrix factorization methods such as *LibFM*. This is probably because neural networks can learn more sophisticated features than *LibFM*, which is beneficial for learning more informative latent factors of users and news.

Second, the methods using negative sampling (*DSSM* and *NPA*) outperform the methods without negative sampling (e.g., *CNN*, *DFM* and *DKN*). This is probably because the methods without negative sampling are usually trained on a balanced dataset with the same number of positive and negative samples, and cannot effectively exploit the rich information conveyed by negative samples. *DSSM*

Table 2: The performance of different methods on news recommendation. *The improvement over all baseline methods is significant at the level $p < 0.001$.

Methods	AUC	MRR	nDCG@5	nDCG@10
LibFM	0.5660	0.2924	0.3015	0.3932
CNN	0.5689	0.2956	0.3043	0.3955
DSSM	0.6009	0.3099	0.3261	0.4185
Wide & Deep	0.5735	0.2989	0.3094	0.3996
DeepFM	0.5774	0.3031	0.3122	0.4019
DFM	0.5860	0.3034	0.3175	0.4067
DKN	0.5869	0.3044	0.3184	0.4071
NPA	**0.6243***	**0.3321***	**0.3535***	**0.4380***

Table 3: The time and memory complexities of different methods. N_s is the number of training samples. D_f and d_f are the dimension of the feature vector and latent factors respectively. *The computational cost per sample.

Methods	Training		Test*	
	Time	Memory	Time	Memory
LibFM	$O(N_s D_f)$	$O(N_s D_f)$	$O(d_f)$	$O(d_f)$
CNN	$O(N_s N D_e)$	$O(N_s N D_e)$	$O(d_f)$	$O(d_f)$
DSSM	$O(N_s D_f / K)$	$O(N_s D_f)$	$O(d_f)$	$O(d_f)$
Wide & Deep	$O(N_s D_f)$	$O(N_s D_f)$	$O(d_f)$	$O(d_f)$
DeepFM	$O(N_s D_f)$	$O(N_s D_f)$	$O(d_f)$	$O(d_f)$
DFM	$O(N_s(ND + D_f))$	$O(N_s(ND + D_f))$	$O(d_f + D_f)$	$O(d_f + D_f)$
DKN	$O(N_s ND)$	$O(N_s ND)$	$O(d_f N)$	$O(ND + d_f)$
NPA	$O(N_s ND/K)$	$O(N_s ND)$	$O(d_f + D + D_e)$	$O(d_f + D_e)$

and our *NPA* approach can utilize the information from three more times of negative samples than other baseline methods, which is more suitable for real-world news recommendation scenarios.

Third, the deep learning methods using attention mechanism (*DFM*, *DKN* and *NPA*) outperform most of the methods without attention mechanism (*CNN*, *Wide & Deep* and *DeepFM*). This is probably because different news and their contexts usually have different informativeness for recommendation, and selecting the important features of news and users is useful for achieving better recommendation performance.

Fourth, our approach can consistently outperform all compared baseline methods. Although *DSSM* also use negative sampling techniques to incorporate more negative samples, it cannot effectively utilize the contextual information and word orders in news titles. Thus, our approach can outperform *DSSM*. In addition, although *DFM* uses attention mechanism to select important user features, it also cannot effectively model the contexts within news titles, and cannot select important words in the candidate news titles. Besides, although *DKN* uses a news-level attention network to select the news clicked by users, it cannot model the informativeness of different words. Different from all these methods, our approach can dynamically recognize important words and news according to user preferences. Thus, our approach can outperform these methods.

Next, we will compare the computational cost of our approach and the baseline methods. To summarize, the comparisons are shown in Table 3[4]. We assume that during the online test phase, the model can directly use the intermediate results produced by hidden layers. From Table 3, we have several observations.

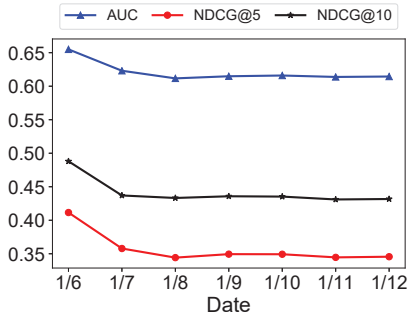


Figure 4: The performance of our *NPA* approach in different days of a week (1/6/2019-1/12/2019).

First, comparing our *NPA* approach with feature-based methods, the computational cost on time and memory during training is lower if N is not large, since the dimension D_f of the feature vector is usually huge due to the dependency on bag-of-words features.[5] In addition, the computational cost in the test phase is only a little more expensive than these methods since D_e is not large.

Second, comparing our *NPA* approach with *CNN*, *DFM* and *DKN*, the training cost is actually divided by K with the help of negative sampling. In addition, the computational cost of *NPA* in the test phase much smaller than *DKN* and *DFM*, since *DKN* needs to use the representations of the candidate news as the query of the news-level attention network and the score needs to be predicted by encoding all news clicked by a user, which is very computationally expensive. *DFM* needs to take the sparse feature vector as input, which is also computationally expensive. Different from these baseline methods, our approach can be trained at a low computational cost, and can be applied to online services to handle massive users at a satisfactory computational cost.

Finally, we want to evaluate the performance of our approach in each day to explore the influence of user click behaviors over time. The performance of our approach in each day of the week for test (1/6/2019-1/12/2019) is shown in Figure 4. According to the results, the performance of our approach is best on the first day in the test week (1/6/2019). This is intuitive because the relevance of user preferences is usually strong between neighbor days. In addition, as time went by, the performance of our approach begins to decline. This is probably because news are usually highly time-sensitive and most articles in common news services will be no longer recommended for users within several days (Usually two days for MSN news). Thus, more news will not appear in the training set over time, which leads to the performance decline. Fortunately, we also find the performance of our approach tends to be stable after three days. It shows that our model does not simply memorize

[4]We omit the length of news titles since it is usually set to a constant. In addition, we also omit the dimensions of hidden layers because they are usually close to the word embedding dimension D.

[5]In practice, the training time of *LibFM* is about 20 minutes, while *DSSM*, *Wide & Deep*, *DeepFM* all take more than one hour on a GTX1080ti GPU. Our *NPA* only takes about 15 minutes.

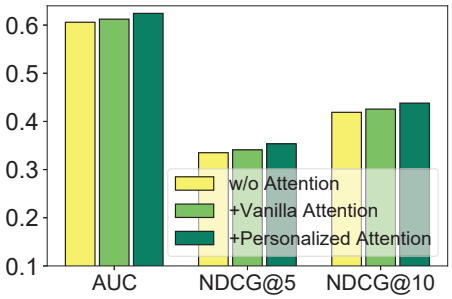


Figure 5: The effectiveness of our personalized attention mechanism compared with vanilla attention.

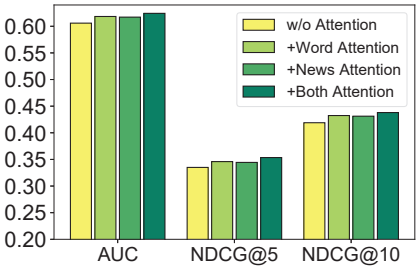


Figure 6: The effectiveness of the word-level and news-level personalized attention network.

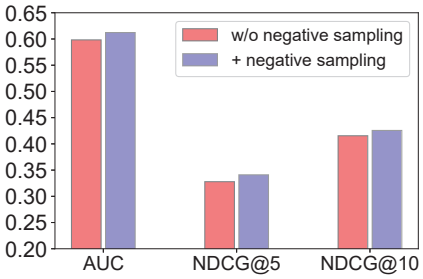


Figure 7: The influence of negative sampling on the performance of our approach.

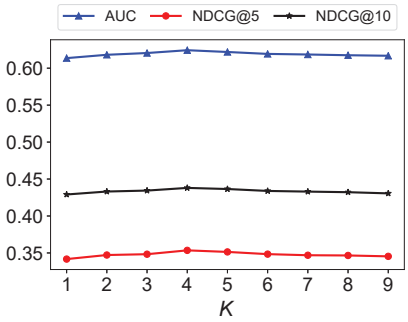


Figure 8: Influence of the negative sampling ratio K on the performance of our approach.

the news appear in the training set and can make personalized recommendations based on user preferences and news topics. Thus, our model is robust to the news update over time.

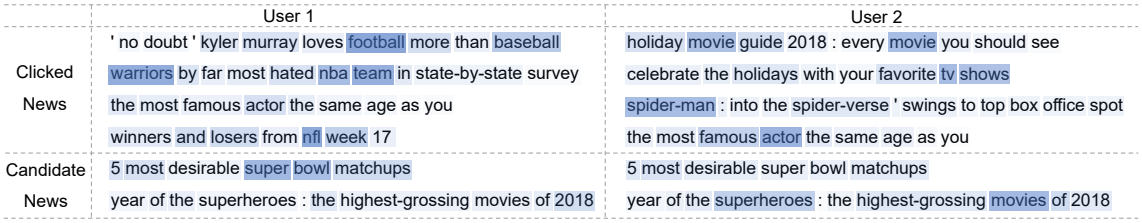
4.3 Effectiveness of Personalized Attention

In this section, we conducted several experiments to explore the effectiveness of the personalized attention mechanism in our *NPA* approach. First, we want to validate the advantage of personalized attention on vanilla non-personalized attention for news recommendation. The performance of *NPA* and its variant using vanilla attention and without attention is shown in Figure 5. According to Figure 5, we have several observations. First, the models with attention mechanism consistently outperform the model without attention. This is probably because different words and news usually have different informativeness for news recommendation. Therefore, using attention mechanism to recognize and highlight important words and news is useful for learning more informative news and user representations. In addition, our model with personalized attention outperforms its variant with vanilla attention. This is probably because the same words and news should have different informativeness for the recommendation of different users. However, vanilla attention networks usually use a fixed query vector, and cannot adjust to different user preferences. Different from vanilla attention, our personalized attention approach can dynamically attend to important words and news according to user characteristics, which can benefit news and user representation learning.

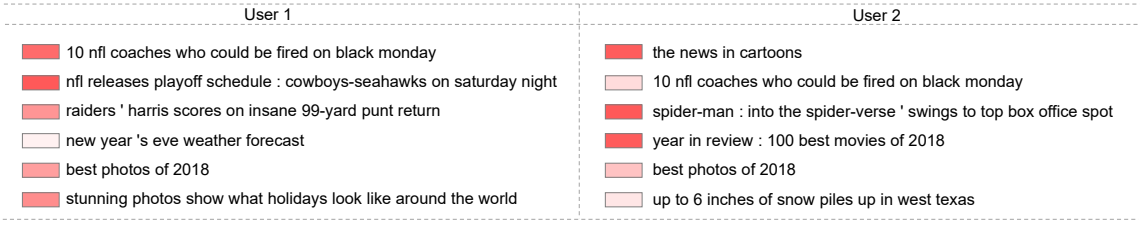
Then, we want to validate the effectiveness of the personalized attention at word-level and news-level. The performance of *NPA* and its variant with different combinations of personalized attention is shown in Figure 6. According to Figure 6, we have several observations. First, the word-level personalized attention can effectively improve the performance of our approach. This is probably because words are basic units to convey the meanings of news titles and selecting the important words according to user preferences is useful for learning more informative news representations for personalized recommendation. Second, the news-level personalized attention can also improve the performance of our approach. This is probably because news clicked by users usually have different informativeness for learning user representations, and recognizing the important news is useful for learning high quality user representations. Third, combining both word- and news-level personalized attention can further improve the performance of our approach. These results validate the effectiveness of the personalized attention mechanism in our approach.

4.4 Influence of Negative Sampling

In this section, we will explore the influence of the negative sampling technique on the performance of our approach. To validate the effectiveness of negative sampling, we compare the performance of our approach with its variant without negative sampling. Following [20, 32], we choose to train this variant on a balanced training set by predicting the click scores of news articles one by



(a) Word-level attention weights.



(b) News-level attention weights.

Figure 9: Visualization of the attention weights from the word- and news-level personalized attention network. The users and news are randomly sampled from the dataset. Darker colors indicate higher attention weights.

one (the final activation function is changed to sigmoid). The experimental results are shown in Figure 7. According to Figure 7, the performance of our approach can be effectively improved via negative sampling. Since negative samples are dominant in the training set, they usually contain rich information for recommendation. However, the information provided by negative samples cannot be effectively utilized due to the limitation of balanced sampling. Therefore, the performance is usually sub-optimal. Different from this variant, our *NPA* approach can incorporate richer information in negative samples, which may be very useful for achieving better performance on news recommendation.

4.5 Hyperparameter Analysis

In this section, we will explore the influence of an important hyperparameter in our approach, i.e., the negative sampling ratio K, which aims to control the number of negative samples to combine with a positive sample. The experimental results on K are shown in Figure 8. According to Figure 8, we find the performance of our approach first consistently improves when K increases. This is probably because when K is too small, the number of negative samples incorporated for training is also small, and the useful information provided by negative samples cannot be fully exploited, which will lead to sub-optimal performance. However, when K is too large, the performance will start to decline. This is probably because when too many negative samples are incorporated, they may become dominant and it is difficult for the model to correctly recognize the positive samples, which will also lead to sub-optimal performance. Thus, a moderate setting of K may be more appropriate (e.g., $K = 4$).

4.6 Case Study

In this section, we will conduct several case studies to visually explore the effectiveness of the personalized attention mechanism in our approach. First, we want to explore the effectiveness of the word-level personalized attention. The visualization results of the clicked and candidate news from two sample users are shown in Figure 9(a). From Figure 9(a), we find the attention network can effectively recognize important words within news titles. For example, the word "nba" in the second news of user 1 is assigned high attention weight since it is very informative for modeling user preferences, while the word "survey" in the same title gains low attention weight since it is not very informative. In addition, our approach can calculate different attention weights for the words in the same news titles to adjust to the preferences of different users. For example, according to the clicked news, user 1 may be interested in sports news and user 2 may be interested in movie related news. The words "super bowl" are highlighted for user 1 and the words "superheroes" and "movies" are highlighted for user 2. These results show that our approach can learn personalized news representations by incorporating personalized attention.

Then, we want to explore the effectiveness of the news-level attention network. The visualization results of the clicked news are shown in Figure 9(b). From Figure 9(b), we find our approach can also effectively recognize important news of a user. For example, the news "nfl releases playoff schedule : cowboys-seahawks on saturday night" gains high attention weight because it is very informative for modeling the preferences of user 1, since he/she is very likely to be interested in sports according to the clicked news. The news "new year's eve weather forecast" is assigned low attention weight, since it is uninformative for modeling user preferences. In addition, our approach can model the different informativeness of news for learning representations of different users. For example, the same sports news "10 nfl coaches who could be fired on black monday" is assigned high attention weight for user 1, but relatively low for user 2. According to the clicked news of both users, user 1 is more likely to be interested in sports than user 2 and this news may be

noisy for user 2. These results show that our model can evaluate the different importance of the same news for different users according to their preferences.

5 CONCLUSION

In this paper, we propose a neural news recommendation approach with personalized attention (NPA). In our *NPA* approach, we use a news representation model to learn news representations from titles using CNN, and use a user representation model to learn representations of users from their clicked news. Since different words and news articles usually have different informativeness for representing news and users, we proposed to apply attention mechanism at both word- and news to help our model to attend to important words and news articles. In addition, since the same words and news usually have different importance for different users, we propose a personalized attention network which exploits the embeddings of user ID as the queries of the word- and news-level attention networks. The experiments on the real-world dataset collected from MSN news validate the effectiveness of our approach.

ACKNOWLEDGMENTS

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. This work was supported by the National Key Research and Development Program of China under Grant number 2018YFC1604002, the National Natural Science Foundation of China under Grant numbers U1836204, U1705261, U1636113, U1536201, and U1536207, and the Tsinghua University Initiative Scientific Research Program under Grant number 20181080368.

REFERENCES

[1] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural News Recommendation with Long- and Short-term User Representations. In *ACL*.
[2] Trapit Bansal, Mrinal Das, and Chiranjib Bhattacharyya. 2015. Content driven user profiling for comment-worthy recommendations of news and blog articles. In *RecSys*. ACM, 195–202.
[3] Michel Capelle, Flavius Frasincar, Marnix Moerland, and Frederik Hogenboom. 2012. Semantics-based news recommendation. In *WIMS*. ACM, 27.
[4] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *WWW*. 1583–1592.
[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *DLRS*. ACM, 7–10.
[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *RecSys*. ACM, 191–198.
[7] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *WWW*. ACM, 271–280.
[8] Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. 2012. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *WSDM*. ACM, 153–162.
[9] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW*. 278–288.
[10] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *AISTATS*. 315–323.
[11] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *AAAI*. AAAI Press, 1725–1731.
[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.
[13] 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*. ACM, 2333–2338.
[14] Wouter IJntema, Frank Goossen, Flavius Frasincar, and Frederik Hogenboom. 2010. Ontology-based news recommendation. In *Proceedings of the 2010 EDBT/ICDT Workshops*. ACM, 16.
[15] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*. 1746–1751.
[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[17] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. 1997. GroupLens: applying collaborative filtering to Usenet news. *Commun. ACM* 40, 3 (1997), 77–87.
[18] 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 (2010), 483–495.
[19] Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. 2011. SCENE: a scalable two-stage personalized news recommendation system. In *SIGIR*. ACM, 125–134.
[20] 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 *IJCAI*. 3805–3811.
[21] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *IUI*. ACM, 31–40.
[22] Zhongqi Lu, Zhicheng Dou, Jianxun Lian, Xing Xie, and Qiang Yang. 2015. Content-Based Collaborative Filtering for News Topic Recommendation. In *AAAI*. 217–223.
[23] Yuanhua Lv, Taesup Moon, Pranam Kolari, Zhaohui Zheng, Xuanhui Wang, and Yi Chang. 2011. Learning to model relatedness for news recommendation. In *WWW*. ACM, 57–66.
[24] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *KDD*. ACM, 1933–1942.
[25] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.
[26] Owen Phelan, Kevin McCarthy, Mike Bennett, and Barry Smyth. 2011. Terms of a feather: Content-based news recommendation and discovery using twitter. In *ECIR*. Springer, 448–459.
[27] Owen Phelan, Kevin McCarthy, and Barry Smyth. 2009. Using twitter to recommend real-time topical news. In *RecSys*. ACM, 385–388.
[28] Steffen Rendle. 2012. Factorization machines with libfm. *TIST* 3, 3 (2012), 57.
[29] Jeong-Woo Son, A Kim, Seong-Bae Park, et al. 2013. A location-based news article recommendation with explicit localized semantic analysis. In *SIGIR*. ACM, 293–302.
[30] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR* 15, 1 (2014), 1929–1958.
[31] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *KDD*. ACM, 1235–1244.
[32] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *WWW*. 1835–1844.
[33] Xuejian Wang, Lantao Yu, Kan Ren, Guanyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In *KDD*. ACM, 2051–2059.
[34] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Attentive Multi-View Learning. In *IJCAI*.
[35] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Topic-Aware News Representation. In *ACL, short paper*.
[36] Chuhan Wu, Fangzhao Wu, Junxin Liu, Shaojian He, Yongfeng Huang, and Xing Xie. 2019. Neural Demographic Prediction using Search Query. In *WSDM*. ACM, 654–662.
[37] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems. In *IJCAI*. 3203–3209.
[38] Shuangfei Zhai, Keng-hao Chang, Ruofei Zhang, and Zhongfei Mark Zhang. 2016. Deepintent: Learning attentions for online advertising with recurrent neural networks. In *KDD*. ACM, 1295–1304.
[39] 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*. 167–176.
[40] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *WSDM*. ACM, 425–434.