

# Denoising Neural Network for News Recommendation with Positive and Negative Implicit Feedback

Yunfan Hu\*

Santa Clara University  
freddy@pku.edu.cn

Zhaopeng Qiu

Tencent  
zhaopengqiu@tencent.com

Xian Wu<sup>†</sup>

Tencent  
kevinxwu@tencent.com

## Abstract

News recommendation is different from movie or e-commercial recommendation as people usually do not grade the news. Therefore, user feedback for news is always implicit (click behavior, reading time, etc). Inevitably, there are noises in implicit feedback. On one hand, the user may exit immediately after clicking the news as he dislikes the news content, leaving the noise in his positive implicit feedback; on the other hand, the user may be recommended multiple interesting news at the same time and only click one of them, producing the noise in his negative implicit feedback. Opposite implicit feedback could construct more integrated user preferences and help each other to minimize the noise influence. Previous works on news recommendation only used positive implicit feedback and suffered from the noise impact. In this paper, we propose a Denoising neural network for news Recommendation with Positive and Negative implicit feedback, named DRPN. DRPN utilizes both feedback for recommendation with a module to denoise both positive and negative implicit feedback to further enhance the performance. Experiments on the real-world large-scale dataset demonstrate the state-of-the-art performance of DRPN.

## 1 Introduction

Online news platforms, such as Google News and Microsoft News, have attracted a large population of users (Wu et al., 2020b). However, massive news articles emerging every day on these platforms make it difficult for users to find appealing content quickly (Wu et al., 2019b). To alleviate the information overload problem, recommender systems have become integral parts of these platforms.

A core problem in news recommendation is how to learn better representations of users and

news (Hu et al., 2020b). Early works include collaborate filtering (CF) based methods (Das et al., 2007), content-based methods (IJntema et al., 2010) and hybrid methods (De Francisci Morales et al., 2012) that combine the two. These methods usually have the cold start problem when being exposed to the sparsity of user-item interactions (Zhu et al., 2019). Recently, deep learning methods have been proposed to learn better user and news representations. The techniques evolve from using recursive neural network (Okura et al., 2017), attention mechanism (Zhu et al., 2019; Wu et al., 2019c), to graph neural network (Wang et al., 2018a; Hu et al., 2020b,a; Qiu et al., 2022). These methods usually recommend news for users based on their historical feedback.

Implicit feedback is more commonly collected than explicit feedback for news because the users usually do not grade the news. Hence, current news recommendation methods naturally use positive implicit feedback like click behavior as the historical feedback to model user interests. However, there are gaps between positive implicit feedback and user real preferences (Wang et al., 2018b). For example, the click behaviors do not fully reflect the user’s preferences. The user may exit the news immediately after clicking, which will involve a noise in the positive feedback. Additionally, some news that users did not click, may also attract them later. Ignoring them also impacts the recommendation performance. Our observation is that using both positive and negative implicit feedback can better model user interests. Besides, positive and negative implicit feedback can help to denoise each other by conducting inter-comparison and intra-comparison. If a news story in one feedback sequence is more similar to the news in the opposite feedback sequence rather than the news in the same sequence, it is very likely that this news story constitutes noise. We can remove this news when building user interests. This idea is shown in Figure 1.

\*This work was done when Yunfan Hu was an intern at Tencent.

<sup>†</sup>Xian Wu is the Corresponding Author.

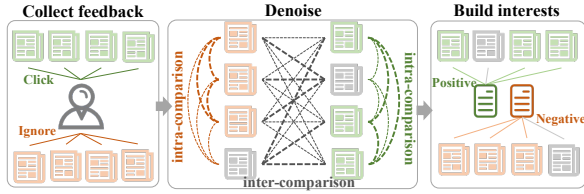


Figure 1: The idea to denoise the implicit feedback. Noises are found by conducting inter-comparison and intra-comparison and then reduced.

In this paper, we propose the **D**enoising neural network for news **R**ecommendation with **P**ositive and **N**egative implicit feedback, named DRPN. It first introduces a news encoder to represent the news in two implicit feedback sequences. Then two parallel aggregators are used to extract user representations from both positive and negative historical feedback: (1) *content-based aggregator*, which selects the informative news in the feedback sequences to represent the user; (2) *denoising aggregator*, which finds and reduces the noises in the feedback sequences. In addition to the semantic information, we introduce a graph neural network to incorporate the collaborative information to further enrich the user representation. Finally, the user and candidate news representations are used to predict the clicking probability. The contributions of this paper are summarized as follows:

- We propose a novel neural news recommendation approach DRPN which jointly models both positive and negative implicit feedback sequences to represent the user to improve recommendation performance.
- In DRPN, to minimize the impacts of the noises in the implicit feedback, the denoising aggregators are designed to refine the two feedback sequences and can help to further improve the recommendation performance.
- The experiments on the large-scale real-world dataset demonstrate that DRPN achieves state-of-the-art performance.

## 2 Related Works

### 2.1 Recommendation with Multi-type Feedback

Few works notice the noise problem in the implicit feedback. (Zhao et al., 2018; Liu et al., 2020) use multiple types of feedback to improve recommendation. However, they ignore the noise in the im-

PLICIT feedback. (Wang et al., 2018b) notices the noise problem but it fails to use the meaningful semantic information in the news. (Wu et al., 2020a; Xie et al., 2020; Bian et al., 2021) use the explicit feedback (such as reading time and like/dislike behaviors) to help denoise the implicit feedback. However, the explicit feedback is harder to collect than the implicit feedback. Differently, DRPN only depends on the implicit feedback (click and non-click behaviors) to conduct the denoise to better model the user preferences.

### 2.2 Graph Neural Network

Recently, graph neural networks (GNN) have received wide attention in many fields (Wu et al., 2020c). The convolutional GNN can learn powerful node representations by aggregating the neighbors’ features. Recently, some works have attempted to leverage the graph information to enhance the representations learning for news recommendation with GNNs. (Wang et al., 2018a) uses entities in news to build a knowledge graph and use the entity embeddings to improve the model performance. (Ge et al., 2020) combines the one- and two-hop neighbor news and users to enrich the representations of the candidate news and user, respectively. However, these methods also depend on the positive implicit feedback to model user representations and ignore the noise problem.

## 3 Problem Formulation

The news recommendation problem in our paper can be illustrated as follows. Let  $\mathcal{U}$  and  $\mathcal{R}$  denote the entire user set and news set. The feedback matrix for the users over the news is denoted as  $\mathcal{Z} \in \mathbb{R}^{l_u \times l_r}$ , where  $z_{u,r} = 1$  means user  $u$  gives a positive implicit feedback to news  $r$  (e.g.,  $u$  clicks  $r$ ),  $z_{u,r} = -1$  means user  $u$  gives a negative implicit feedback to news  $r$  (e.g.,  $u$  sees  $r$  but ignores it), and  $z_{u,r} = 0$  means no feedback.  $l_u$  and  $l_r$  denote the numbers of the users and news, respectively. For each specific user, his historical positive feedback sequence  $[p_1, \dots, p_{l_p}]$  and negative feedback sequence  $[n_1, \dots, n_{l_n}]$  can be gathered from the feedback matrix  $\mathcal{Z}$ , where  $p_i, n_j \in \mathcal{R}$ .

Given the feedback matrix  $\mathcal{Z}$ , the goal is to train a model  $\mathcal{M}$  (i.e., DRPN). For each new pair of user and candidate news ( $u \in \mathcal{U}, r \in \mathcal{R}$ ), we can use  $\mathcal{M}$  to estimate the probability that  $u$  would like to click  $r$ .

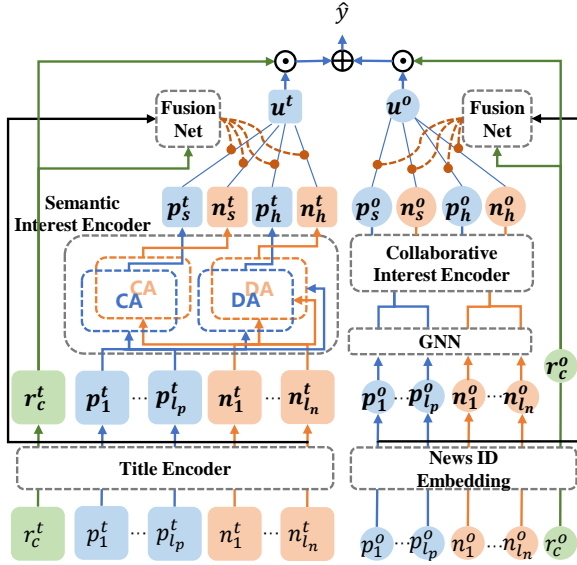


Figure 2: The DRPN framework.

## 4 Framework

Figure 2 shows the architecture of DRPN. It first employs the title encoder and id embedding layer to represent all news in two feedback sequences and the candidate news. Then two separate encoders are employed to extract the user semantic interest and collaborative interest information from both positive and negative implicit feedback sequences. Next, two fusion nets combine multiple interest representations to represent the user. Finally, we use the user and candidate news representations to estimate the clicking probability. We will detail each component in the following subsections.

### 4.1 Input

The inputs of the DRPN contain six parts: the titles of positive feedback sequence  $[p_1^t, \dots, p_{l_p}^t]$ , the titles of negative feedback sequence  $[n_1^t, \dots, n_{l_n}^t]$ , the candidate news title  $r_c^t$ , the IDs of positive feedback sequence  $[p_1^o, \dots, p_{l_p}^o]$ , the IDs of negative feedback sequence  $[n_1^o, \dots, n_{l_n}^o]$ , and candidate news ID  $r_c^o$ .

For each news title  $t$ , we convert its every word  $w$  to a  $d$ -dimensional vector  $\mathbf{w}$  via an embedding matrix  $\mathbf{E}_W \in \mathbb{R}^{l_w \times d}$ , where  $l_w$  is the vocabulary size and  $d$  is the dimension of word embedding. Then, the title  $t$  is transformed into a matrix  $\mathbf{T}$ .

For each news ID  $o$ , we also convert it to a  $d$ -dimensional vector  $\mathbf{o}$  via an embedding matrix  $\mathbf{E}_O \in \mathbb{R}^{l_r \times d}$ . In this manner, we can encode all news to obtain  $\mathbf{P}^o = [p_1^o, \dots, p_{l_p}^o]$  ( $p_i^o \in \mathbb{R}^d$ ),  $\mathbf{N}^o = [n_1^o, \dots, n_{l_n}^o]$  ( $n_i^o \in \mathbb{R}^d$ ), and  $r_c^o \in \mathbb{R}^d$ .

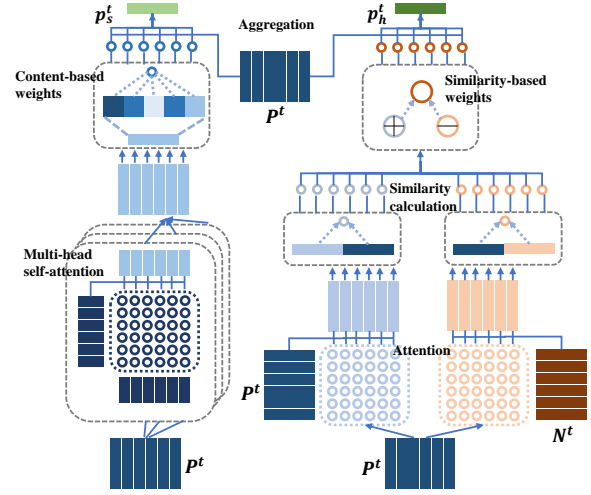


Figure 3: The content-based aggregator (CA) on the left and the denoising aggregator (DA) on the right in semantic interest encoder. They encode positive preferences for the user. (*best viewed in color*)

### 4.2 Title Encoder

The title encoder can extract the sentence-level semantic representation of the news title. It contains two sub-layers. We take the title  $\mathbf{T}$  as an example to detail the encoding process.

The first sub-layer is a multi-head self-attention layer, which can model the contextual representation of each word. Given three input matrices  $\mathbf{Q} \in \mathbb{R}^{l_q \times d}$ ,  $\mathbf{K} \in \mathbb{R}^{l_v \times d}$  and  $\mathbf{V} \in \mathbb{R}^{l_v \times d}$ , the attention function is defined as:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\mathbf{Q}\mathbf{K}^\top / \sqrt{d})\mathbf{V} \quad (1)$$

Multi-head self-attention layer  $\text{MH}(\cdot, \cdot, \cdot)$  will further project the input to multiple semantic subspaces and capture the interaction information from multiple views:

$$\text{MH}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\text{head}_1; \dots; \text{head}_{l_h}]\mathbf{W}^I \quad (2)$$

$$\text{head}_i = \text{Attn}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$

where  $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d \times d/l_h}$  and  $\mathbf{W}^I \in \mathbb{R}^{d \times d}$  are the parameters to learn.  $l_h$  is the number of heads.  $[\cdot]$  denotes the column-wise concatenation for matrices.

Moreover, we employ the residual connection and layer normalization function  $\text{LN}$  defined in (Ba et al., 2016) to fuse the original and contextual representations:  $\tilde{\mathbf{T}} = \text{LN}(\mathbf{T} + \text{MH}(\mathbf{T}, \mathbf{T}, \mathbf{T}))$ .

The second sub-layer is a gated aggregation layer (Qiu et al., 2020). It will select the important words to generate an informative title representation. The gated mechanism is employed to

decide the weight of each words. Given the word embedding matrix  $\tilde{\mathbf{T}}$ , its sentence-level semantic representation  $\mathbf{t}$  is calculated as follows:

$$\begin{aligned} \mathbf{t} &= \text{Aggregate}(\tilde{\mathbf{T}}) \\ &= \left( \text{softmax}(\tanh(\tilde{\mathbf{T}}\mathbf{W}_a + \mathbf{b}_a)\mathbf{W}_g) \right)^\top \tilde{\mathbf{T}} \quad (3) \end{aligned}$$

where  $\mathbf{W}_a \in \mathbb{R}^{d \times d'}$ ,  $\mathbf{b}_a \in \mathbb{R}^{d'}$  and  $\mathbf{W}_g \in \mathbb{R}^{d' \times 1}$  are trainable parameters.

Finally, we can use the title encoder to model the titles of all news in two user feedback sequences to obtain  $\mathbf{P}^t = [\mathbf{p}_1^t, \dots, \mathbf{p}_{l_p}^t]$  and  $\mathbf{N}^t = [\mathbf{n}_1^t, \dots, \mathbf{n}_{l_n}^t]$ . For the candidate news, we can also obtain its title representation  $\mathbf{r}_c^t$  via the same title encoder.

### 4.3 Semantic Interest Encoder

The titles of the news which the user interacted usually reflect the user's interests. Hence, we can learn user interest representations by encoding the semantic representations of the news. As is shown in Figure 3, the semantic interest encoder leverages two aggregators, *content-based aggregator* (CA) and *denoising aggregator* (DA), to extract user preferences from both positive and negative feedback sequences.

#### 4.3.1 Content-based Aggregator

Different news have different informativeness when representing users. For example, sport news are more informative than weather news in modeling user personality, since the latter are usually browsed by most users. The content-based aggregator (CA) will first evaluate the importance of different news in the feedback sequence from the content view and then aggregate the important news to represent the user. It contains two sub-layers.

The first one is a multi-head self-attention layer, which can enhance the news representations by capturing their interactions. For the positive feedback sequence  $\mathbf{P}^t$ , the multi-head self-attention layer generates  $\tilde{\mathbf{P}}^t = \text{LN}(\mathbf{P}^t + \text{MH}(\mathbf{P}^t, \mathbf{P}^t, \mathbf{P}^t))$ . The MH is define in Eq.(2) with independent parameters and the LN is the layer normalization function.

The second sub-layer is a gated aggregation layer that has the same structure as the one defined in Eq.(3). For  $\tilde{\mathbf{P}}^t$ , it can select the more informative news to generate the user representation:  $\mathbf{p}_s^t = \text{Aggregate}(\tilde{\mathbf{P}}^t)$ . We also use the content-based aggregator to generate another user representation from the negative feedback sequence  $\mathbf{N}^t$ ,  $\mathbf{n}_s^t$ .

#### 4.3.2 Denoising Aggregator

Denoising aggregator will conduct what we call a refining operation, which aims to mitigate the impacts of the noises in the feedback when modeling the user interests. Intuitively, if news clicked by the user is more semantically relevant with the news in the positive feedback sequence, this news is more likely the user true preference. Otherwise, if it is more semantically relevant with the news in the negative feedback sequence, it is more likely a noise for representing the user interest. As shown in Figure 3, for each news in the positive feedback sequence, we will conduct the intra-comparisons with the news in the positive sequence and inter-comparisons with the news in the negative sequence to decide its weight when representing the user. This module contains three sub-layers.

The first sub-layer is an intra-attention layer. For news  $\mathbf{p}_j^t \in \mathbf{P}^t$ , this layer uses it as the query to aggregate all news in  $\mathbf{P}^t$  except  $\mathbf{p}_j^t$  by the attention mechanism to obtain the sequence-level representation,  $\hat{\mathbf{p}}_j^t = (\sum_{i \neq j} \alpha_{ji}^h \mathbf{p}_i^t) \mathbf{W}_v^h$ , where

$$\alpha_{ji}^h = \frac{\exp\left(\frac{(\mathbf{p}_j^t \mathbf{W}_q^h)(\mathbf{p}_i^t \mathbf{W}_k^h)^\top}{\sqrt{d}}\right)}{\sum_{e \neq j} \exp\left(\frac{(\mathbf{p}_j^t \mathbf{W}_q^h)(\mathbf{p}_e^t \mathbf{W}_k^h)^\top}{\sqrt{d}}\right)} \quad (4)$$

$\mathbf{W}_q^h, \mathbf{W}_k^h, \mathbf{W}_v^h \in \mathbb{R}^{d \times d}$  are learnable parameters.

The second sub-layer is an inter-attention layer. For  $\mathbf{p}_j^t$ , this layer uses it as the query to aggregate its relevant news in the negative feedback sequence  $\mathbf{N}^t$  by the attention mechanism.

$$\hat{\mathbf{n}}_j^t = \text{Attn}(\mathbf{p}_j^t \mathbf{W}_q^h, \mathbf{N}^t \mathbf{W}_k^{h'}, \mathbf{N}^t \mathbf{W}_v^{h'}) \quad (5)$$

where  $\mathbf{W}_k^{h'}, \mathbf{W}_v^{h'} \in \mathbb{R}^{d \times d}$  are learnable parameters.

The third sub-layer is a gated aggregation layer. The weight of the news  $\mathbf{p}_j^t$  are decided by the semantic similarities between  $\mathbf{p}_j^t$  and two sequence-level representations,  $\hat{\mathbf{p}}_j^t$  and  $\hat{\mathbf{n}}_j^t$ .

$$\begin{aligned} s_j^p &= \tanh([\mathbf{p}_j^t; \hat{\mathbf{p}}_j^t] \mathbf{W}_1^u + \mathbf{b}_1^u) \mathbf{W}_2^u + b_2^u \\ s_j^n &= \tanh([\mathbf{p}_j^t; \hat{\mathbf{n}}_j^t] \mathbf{W}_3^u + \mathbf{b}_3^u) \mathbf{W}_4^u + b_4^u \\ \alpha_j &= \frac{\exp(s_j^p - \text{ReLU}(\gamma) * s_j^n)}{\sum_{i=0}^{i=l_p} \exp(s_i^p - \text{ReLU}(\gamma) * s_i^n)} \end{aligned} \quad (6)$$

where  $\mathbf{W}_1^u, \mathbf{W}_3^u \in \mathbb{R}^{2d \times d}$ ,  $\mathbf{W}_2^u, \mathbf{W}_4^u \in \mathbb{R}^{d \times 1}$ ,  $\mathbf{b}_1^u, \mathbf{b}_3^u \in \mathbb{R}^d$ ,  $b_2^u, b_4^u \in \mathbb{R}$  and  $\gamma$  are learnable parameters. Then, this layer will aggregate all news

according to their weights to obtain the denoised representation,  $\mathbf{p}_h^t = \sum_{j=0}^{j=l_p} \alpha_j \mathbf{p}_j^t$ .

For the negative feedback sequence  $\mathbf{N}^t$ , we take a dual denoising process to obtain its final representation  $\mathbf{n}_h^t$ .

#### 4.4 Graph Neural Network

If two news,  $r_i$  and  $r_j$ , are co-clicked by the user  $u_1$  and  $r_i$  is also clicked by  $u_2$ ,  $u_2$  may also prefers  $r_j$  based on the idea of the collaborative filtering. Hence, we can further enrich the user interest representations by modeling the collaborative information. Like the knowledge graph, we build a collaborative graph  $\mathcal{G} = \{(r_i, r_j) | r_i, r_j \in \mathcal{R}\}$  over the news set  $\mathcal{R}$  based on the co-clicking relationships in the historical feedback matrix  $\mathcal{Z}$ .  $(r_i, r_j)$  indicates they are neighbors in the graph and have been clicked by the same user. To incorporate the collaborative information, we employ the graph transformer neural network (Shi et al., 2021) to model the news in the user feedback sequence.

First, for each news node  $\mathbf{r}^o$  in  $\mathbf{P}^o$  and  $\mathbf{N}^o$ , we compute the attention weights between it and its neighbors  $\mathcal{N}(\mathbf{r}^o)$  in  $\mathcal{G}$ .  $\mathcal{N}(\mathbf{r}^o)$  denotes the neighbor set of node  $\mathbf{r}^o$ . Take its neighbor  $\mathbf{r}_k^o$  ( $k \in \mathcal{N}(\mathbf{r}^o)$ ) as an example, the attention weight between  $\mathbf{r}^o$  and  $\mathbf{r}_k^o$  at the  $m$ -th head is calculated by

$$\alpha_k^m = \frac{\exp\left(\left(\mathbf{r}^o \mathbf{W}_{m,1}^g\right)\left(\mathbf{r}_k^o \mathbf{W}_{m,2}^g\right)^\top / \sqrt{\hat{d}}\right)}{\sum_{q \in \mathcal{N}(\mathbf{r}^o)} \exp\left(\left(\mathbf{r}^o \mathbf{W}_{m,1}^g\right)\left(\mathbf{r}_q^o \mathbf{W}_{m,2}^g\right)^\top / \sqrt{\hat{d}}\right)}$$

where  $\mathbf{W}_{m,*}^g \in \mathbb{R}^{d \times d/l'_h}$  are learnable parameters.  $l'_h$  is the number of heads and  $\hat{d}$  is equal to  $d/l'_h$ .

Next, each news node will aggregate the information of its neighbors from multiple heads according to the attention weights. For the node  $\mathbf{r}^o$ , the representation aggregated from its neighbors is:

$$\hat{\mathbf{r}}^o = \left\| \sum_{m=1}^{m=l'_h} \left\{ \sum_{k \in \mathcal{N}(\mathbf{r}^o)} \alpha_k^m \mathbf{r}_k^o \mathbf{W}_{m,3}^g \right\} \right\| \quad (7)$$

where  $\{\mathbf{W}_{m,3}^g \in \mathbb{R}^{d \times d/l'_h}\}_{m=1}^{m=l'_h}$  are trainable parameters.  $\|$  denotes the concatenation operation for  $l'_h$  heads.

Finally, we will update the representation of each node by fusing its aggregated and original representations.

$$\begin{aligned} \tilde{\mathbf{r}}^o &= \text{Fuse}(\mathbf{r}^o, \hat{\mathbf{r}}^o) \\ &= \sigma([\mathbf{r}^o; \hat{\mathbf{r}}^o] \mathbf{W}_1^f) \odot \tanh([\mathbf{r}^o; \hat{\mathbf{r}}^o] \mathbf{W}_2^f) \end{aligned} \quad (8)$$

where  $\mathbf{W}_1^f, \mathbf{W}_2^f \in \mathbb{R}^{2d \times d}$  are learnable parameters.  $\odot$  denotes the element-wise multiplication operation.  $\sigma$  is the sigmoid function.

We can use this graph neural network to encode all news in the user positive and negative feedback sequences to obtain  $\tilde{\mathbf{P}}^o = [\tilde{\mathbf{p}}_1^o, \dots, \tilde{\mathbf{p}}_{l_p}^o]$  and  $\tilde{\mathbf{N}}^o = [\tilde{\mathbf{n}}_1^o, \dots, \tilde{\mathbf{n}}_{l_n}^o]$ .

#### 4.5 Collaborative Interest Encoder

The module aims to model user interests by aggregating the representations of two feedback sequences encoded by the graph neural network layer, which have incorporated the collaborative information. The structure of the collaborative interest encoder is similar to that of the semantic interest encoder and also contains two aggregators, a content-based aggregator and a denoising aggregator. The denoising aggregators have the same structure as the one in the semantic interest encoder. The only structural difference between two content-based aggregators of two encoders is that there is no multi-head self-attention operation in the content-based aggregator of the collaborative interest encoder. This is because the context information is already propagated by the graph neural network, which has a similar effect with the multi-head self-attention.

The inputs of this encoder are the positive sequence representation  $\tilde{\mathbf{P}}^o$  and the negative sequence representation  $\tilde{\mathbf{N}}^o$ . The content-based aggregator will generate two user representations,  $\mathbf{p}_s^o$  and  $\mathbf{n}_s^o$ , based on two sequence representations, respectively. Similarly, the denoising aggregator will denoise two sequences and generate two user representations  $\mathbf{p}_h^o$  and  $\mathbf{n}_h^o$ .

#### 4.6 Fusion Net

There are two fusion nets as shown in Figure 2. They are used to fuse multiple user interest representations extracted by two interest encoders to form a comprehensive user representation. For different user-candidate news  $(u, r)$  pairs, the fusion net dynamically allocates different weights for different interest representations. Two fusion nets have similar structures but different parameters. We take the one for the semantic interest encoder as an example to detail the fusion process.

The fusion net first represents the  $(u, r)$  pair. It should mitigate the effect of two interest encoders and independently calculate the weights for the output representations of two encoders. Hence, it uses the outputs of the title encoder

to represent  $(u, r)$ ,  $\mathbf{f}^t = [\mathbf{u}_f^t; \mathbf{r}_c^t]$ , where  $\mathbf{u}_f^t = \text{Aggregate}([\mathbf{P}^t | \mathbf{N}^t])$ .  $\mathbf{P}^t$ ,  $\mathbf{N}^t$  and  $\mathbf{r}_c^t$  are the title representations of the news in user positive and negative feedback sequences and the candidate news extracted by the title encoder.  $[[]]$  denotes the row-wise concatenation for matrices.

Then, this module leverages four different fully connected layers to calculate the weights for four representations extracted by the semantic interest encoder (i.e.,  $\mathbf{p}_s^t$ ,  $\mathbf{n}_s^t$ ,  $\mathbf{p}_h^t$  and  $\mathbf{n}_h^t$ ). For example, the weight of  $\mathbf{p}_s^t$  is calculated by

$$\beta_s^p = \tanh(\mathbf{f}^t \mathbf{W}_1^{p,s} + \mathbf{b}_1^{p,s}) \mathbf{W}_2^{p,s} + b_2^{p,s} \quad (9)$$

where  $\mathbf{W}_1^{p,s} \in \mathbb{R}^{2d \times d}$ ,  $\mathbf{W}_2^{p,s} \in \mathbb{R}^{d \times 1}$ ,  $\mathbf{b}_1^{p,s} \in \mathbb{R}^d$ ,  $b_2^{p,s} \in \mathbb{R}$  are learnable parameters. The weights,  $\beta_s^n$ ,  $\beta_h^p$  and  $\beta_h^n$ , of the representations  $\mathbf{n}_s^t$ ,  $\mathbf{p}_h^t$  and  $\mathbf{n}_h^t$  can be calculated by the same way in Eq.(9).

Finally, the user content-view representation is calculated by

$$\mathbf{u}^t = \beta_s^p * \mathbf{p}_s^t + \beta_s^n * \mathbf{n}_s^t + \beta_h^p * \mathbf{p}_h^t + \beta_h^n * \mathbf{n}_h^t \quad (10)$$

Another fusion net is used to fuse four interest representations extracted by the collaborative interest encoder and has a similar structure with the above one. The only difference is that it uses the outputs of the news ID embedding layer to represent the  $(u, r)$  pair,  $\mathbf{f}^o = [\mathbf{u}_f^o; \mathbf{r}_c^o]$ , where  $\mathbf{u}_f^o = \text{Aggregate}([\mathbf{P}^o | \mathbf{N}^o])$ . Then, it uses  $\mathbf{f}^o$  to calculate the respective weights  $\theta_s^p$ ,  $\theta_s^n$ ,  $\theta_h^p$  and  $\theta_h^n$  for four interest representations  $\mathbf{p}_s^o$ ,  $\mathbf{n}_s^o$ ,  $\mathbf{p}_h^o$  and  $\mathbf{n}_h^o$  by the same way in Eq.(9). The final user graph-view representation is calculated by

$$\mathbf{u}^o = \theta_s^p * \mathbf{p}_s^o + \theta_s^n * \mathbf{n}_s^o + \theta_h^p * \mathbf{p}_h^o + \theta_h^n * \mathbf{n}_h^o \quad (11)$$

## 4.7 Prediction

Following (Wu et al., 2019c), the clicking probability score  $\hat{y}$  is computed by the inner product of the user representation and the candidate news representation:  $\hat{y} = \mathbf{u}^t \top \mathbf{r}_c^t + \mathbf{u}^o \top \mathbf{r}_c^o$ , where  $\mathbf{u}^t \top \mathbf{r}_c^t$  stands for the score calculated from title information and  $\mathbf{u}^o \top \mathbf{r}_c^o$  stands for the score calculated from collaborative information.

## 4.8 Training

Following (Wu et al., 2019c), for each positive sample, we randomly select  $l_k$  negative samples from the same user to construct a  $l_k + 1$  classification task. Each output of the DRPN for a classification sample is like  $[\hat{y}^+, \hat{y}_1^-, \dots, \hat{y}_{l_k}^-]$ , where  $\hat{y}^+$  denotes

# users	654870	# avg. titles words	12.66
# news	104153	# positive samples	1048414
# words	54292	# negative samples	25145229
# avg. positive feedback sequence length			4.14
# avg. negative feedback sequence length			96.80

Table 1: Statistics of the dataset.

the clicking probability score of the positive sample and the rest denote the scores of the  $l_k$  negative samples. We define the training loss (to be minimized) as follows.

$$\mathcal{L} = - \sum_{i \in \mathcal{P}} \log \left( \frac{\exp(\hat{y}_i^+)}{\exp(\hat{y}_i^+) + \sum_{j=1}^{j=l_k} \exp(\hat{y}_{i,j}^-)} \right) \quad (12)$$

where  $\mathcal{P}$  denotes the set of positive samples.

## 4.9 Computation Complexity

The time complexity of the title encoder is  $\mathcal{O}(L^2d + Ld^2)$ , where  $L$  is the title length and  $d$  is the embedding size. The time complexity of each interest encoder is  $\mathcal{O}((l_p + l_n)d^2 + (l_p^2 + l_n^2 + (l_p + l_n)^2)d)$  where  $l_p$  and  $l_n$  are the lengths of positive and negative feedback sequences. The time complexity of GNN is  $\mathcal{O}(|\mathcal{G}|d)$ , where  $|\mathcal{G}|$  denotes the number of edges that existed in collaborative graph. Hence, The overall time cost is  $\mathcal{O}((l_p + l_n)(Ld^2 + L^2d) + (l_p^2 + l_n^2 + (l_p + l_n)^2 + |\mathcal{G}|)d)$ .

During the inference phase, we can compute the news representations in advance and the computation complexity will be  $\mathcal{O}((l_p^2 + l_n^2 + (l_p + l_n)^2)d)$ .

## 5 Experiment

### 5.1 Dataset

There is no off-the-shelf dataset in which the user profile includes both positive and negative historical feedback sequences. Therefore, we use MIND<sup>1</sup> dataset (its original user profile only contains positive feedback) to re-build one to conduct the experiments. The original MIND dataset contains the user impression logs. An impression log records the news displayed to a user when visiting the news website homepage at a specific time, and the click behaviors on the news list. We re-build the dataset based on the MIND’s impression logs as follows: (1) Select the impression logs of the first 5 days of the original training set. Then we add the news that a user has seen but did not click to his negative feedback sequence, and add the news he clicked

<sup>1</sup><https://msnews.github.io/>

to his positive feedback sequence. In this manner, the user profile includes both positive and negative historical feedback sequences; (2) Training set: the impression logs of 6-th day of the original training set; (3) Validation set: the first 10% chronological impression logs of the original validation set; (4) Testing set: the last 90% chronological impression logs of the original validation set.

The training, validation, and testing sets use the same user profile built in Step (1). Since the user profiles are only built in Step (1) which is ahead of Step (2)-(4), there is no label leakage to validation and testing sets. Moreover, same as the original MIND dataset, the re-built dataset also has 44.6% users of validation set and 48.7% users of test set that are not shown in the re-built training set. Table 1 shows some statistics of the re-built dataset.

## 5.2 Baseline Approaches and Metrics

We evaluate the performance of DRPN by comparing it with several baseline methods, including: (1) *LibFM* (Rendle, 2012), factorization machine (FM); (2) *DeepFM* (Guo et al., 2017), which combines the FM and neural networks; (3) *DKN* (Wang et al., 2018a), which uses the CNN to fuse the entity and word embeddings to learn news representations; (4) *LSTUR* (An et al., 2019), which uses the GRU to model short- and long-term interests from the click history; (5) *NPA* (Wu et al., 2019b), which introduces the attention mechanism to select important words and news; (6) *DEERS* (Zhao et al., 2018), which uses GRU to encode positive and negative feedback sequences; (7) *DFN* (Xie et al., 2020), a factorization-machine based network which uses transformers to encode both positive and negative feedback sequences to enhance performance; (8) *GERL* (Ge et al., 2020), which constructs user-news graph to enhance the performance; (9) *NAML* (Wu et al., 2019a), which uses multi-view learning to aggregate different kinds of information to represent news; (10) *NRMS* (Wu et al., 2019c), which uses multi-head self-attention to learn news and user representations; (11) *NAML + TCE*, which incorporates the denoising training strategy TCE (Wang et al., 2021) into NAML; (12) *NRMS + TCE*, which improves NRMS by using TCE.

Following the previous news recommendation work (Wu et al., 2020b, 2019c; Ge et al., 2020), we use AUC, MRR, nDCG@5, and nDCG@10 scores as our evaluation metrics.

## 5.3 Implementation Details

For DRPN, the representation dimension  $d$  is set to 300. We use the *GloVe.840B.300d* (Pennington et al., 2014) as the pre-trained word embeddings. The maximum title length is set to 15. The lengths of feedback sequences  $l_p$  and  $l_n$  are set to 30 and 60. Padding and truncation are used to keep sequence and word numbers the same. The head number  $l_h$  in multi-head self-attention is set to 6. The hidden size  $d'$  in the gated aggregation layer is set to 200. The head number in graph neural network  $l'_h$  is set to 2. The negative sampling ratio  $l_k$  is set to 4. When preparing data for graph neural network, we only input sub-graph that contains nodes in the user feedback sequences. Moreover, we pick the maximum 5 neighbor nodes for each node  $r$  in user feedback sequences, which are most frequently co-clicked with  $r$ . We have also released the source code at <https://github.com/chungdz/DRPN>.

For NRMS, DKN, LSTUR, NPA, and NAML, we use the official code and settings<sup>2</sup>. For others, we reimplement them and set their parameters based on the experimental setting strategies reported by their papers.

For fair comparisons, all methods only use the news ID, title, category and subcategory as features. The validation set was used for tuning hyperparameters and the final performance comparison was conducted on the test set.

Models	AUC	MRR	nDCG@5	nDCG@10
LibFM	60.48	26.38	27.75	34.63
DeepFM	62.18	27.26	29.08	35.68
DKN	64.00	28.98	31.49	38.22
LSTUR	65.31	30.31	33.34	39.86
NPA	64.35	29.61	32.88	39.23
DEERS	65.29	30.78	33.78	40.09
DFN	63.11	29.14	31.88	38.33
GERL	64.08	29.34	32.50	38.96
NAML	65.84	30.60	33.89	40.23
NRMS	65.46	30.73	33.78	40.13
NAML + TCE	65.95	30.66	33.93	40.52
NRMS + TCE	65.84	31.58	34.93	41.26
<b>DRPN</b>	<b>67.30</b>	<b>32.68</b>	<b>36.27</b>	<b>42.33</b>

Table 2: Performance comparison of all methods. Best results are highlighted in bold.

## 5.4 Performance Evaluation

The experimental results of all models are summarized in Table 2. We make the following observations from the results. First, our proposed

<sup>2</sup><https://github.com/microsoft/recommenders>

Positive implicit feedback			Negative implicit feedback		
Weights	Categories	Titles	Weights	Categories	Titles
■	<b>Sports</b>	The Geno Smith ... thing in Seahawks' win over 49ers.	■	<b>Finance</b>	Confidence in the US economy accelerates.
■	<b>Sports</b>	Russell Wilson has MVP ... beat 49ers in OT classic.	■	<b>Music</b>	Top 100 country songs of all time.
■	<b>Weather</b>	... farmers endure major crop and profit losses as climate changes.	■	<b>Lifestyle</b>	200 shocking home photos you have to see.
■	<b>Sports</b>	Russell Wilson, Richard Sherman swap jerseys despite ...	■	<b>Politics</b>	New poll shows Buttigieg in the lead in Iowa. Can he win it all?
■	...		■	...	
■	<b>Movies</b>	Actress accuses Roman Polanski of raping her in 1975.	■	<b>Sports</b>	Watch: John Harbaugh shared awesome moment with Lamar Jackson.
■	<b>Finance</b>	Dean Foods files for bankruptcy.	■	<b>Sports</b>	Angels hire Tony La Russa as baseball operations special assistant.
■	<b>Music</b>	Broadway actress Laurel Griggs dies at Age 13.	■	<b>Sports</b>	Fred Taylor on Pro Football Hall of Fame: 'I think I belong'.
■	<b>Lifestyle</b>	A master suite ... is asking for \$1,200/month in rent.	■	<b>Sports</b>	What this week's 5 biggest upsets mean for the NFL playoff picture.
Top 2 recommended news			Last 2 recommended news		
Feedbacks	Categories	Titles	Feedbacks	Categories	Titles
<i>Clicked</i>	<b>Sports</b>	Colin Kaepernick is about to get what he deserves: a chance.	<i>Ignored</i>	<b>Moives</b>	... Disney for casting 'the White Guy' in The Little Mermaid.
<i>Clicked</i>	<b>Sports</b>	... after Redskins' player Montae Nicholson took her to hospital.	<i>Ignored</i>	<b>Music</b>	Taylor Swift ... be owed \$7.9 million in unpaid royalties.

Figure 4: Visualization of the attention weights for an example user’s feedback in the denoising aggregator and the recommendation results for him in validation dataset. A darker color indicates a larger attention weight.

Models	AUC	MRR	nDCG@5	nDCG@10
<b>DRPN</b>	<b>67.30</b>	<b>32.68</b>	<b>36.27</b>	<b>42.33</b>
DRPN-D	66.51	31.09	35.40	41.63
DRPN-G	66.82	31.90	35.34	41.62
DRPN-DG	66.14	31.17	34.57	40.97
DRPN-N	66.38	31.39	34.79	41.13
DRPN-P	65.89	31.11	34.38	40.59

Table 3: Performance comparison of all variants of DRPN. Best results are highlighted in bold.

model, DRPN, outperforms all baselines on the news recommendation datasets. Second, among all baselines, the methods which use the deep neural networks to model the news (i.e., DKN, NPA, LSTUR, DFN, DEERS, NAML, GERL, NRMS) perform better than the feature-based methods (e.g., LibFM and DeepFM). This performance improvement should be attributed to better news representation methods. Among the deep neural methods, NRMS+TCE achieves the best performance by using two level multi-head self-attention to learn user representations and using TCE to denoise the negative samples. Third, among two baselines that use both positive and negative feedback, DFN performs worse than DEERS. The reason may be that original DFN depends on the explicit feedback but the experimental dataset only contains implicit feedback. Compared with NAML, DEERS has a competitive performance even if its news encoder is a simple pooling layer. This also proves the effectiveness of the negative implicit feedback.

## 5.5 Ablation Study

To highlight the individual contribution of each module, we use the following variants of DRPN to run an ablation study: (1) *DRPN-D*, which removes the denoising aggregator; (2) *DRPN-G*, which re-

moves the knowledge graph part; (3) *DRPN-DG*, which removes the knowledge graph part and the denoising aggregator; (4) *DRPN-N*, which only uses the positive feedback; (5) *DRPN-P*, which only uses the negative feedback.

The results are shown in Table 3. First, DRPN-D and DRPN-G perform worse than DRPN, proving the effectiveness of the designed denoising module and the collaborative graph. Second, the results of DRPN-N and DRPN-P indicate the effectiveness of negative and positive feedback, respectively. Third, even without deliberately designing, by using both positive and negative implicit feedback, DRPN-DG can achieve competitive performance compared with the strongest baseline NRMS+TCE. This further proves the effectiveness of the negative feedback.

## 5.6 Case Study

To intuitively illustrate the effectiveness of the denoising aggregator, we sample a user and visualize his historical feedback attention weights in the denoising aggregator of the semantic interest encoder. The upper part of the Figure 4 shows the attention weights and ranks the news in descending order of the attention weight. We can find in positive feedback sequence, the top 4 news are about sports and weather and the last 4 news are about music, movie, finance, and lifestyle. Meanwhile, in negative feedback sequence, the top 4 news are about finance, music, politics, and lifestyle, and the last 4 news are all about sports. This indicates that the denoising aggregator believes that the user likes sports, and dislikes the topics such as finance, music, movies, politics, and lifestyle. As shown in the lower part of Figure 4, based on the predicted user preferences, we can see DRPN prefers to recommend the sports news for this user. Moreover, in the validation data,



we can observe that this user clicks the top 2 recommended news and ignores the last 2 news. It suggests the user preference extracted by the denoising aggregator is consistent with the user's real behaviors. In summary, the visualization results indicate the denoising module can better capture the user's real preferences by conducting the inter- and intra- comparisons between the positive and negative implicit feedback sequences.

## 6 Conclusion

In this paper, we propose a novel deep neural news recommendation model DRPN. In DRPN, we design two aggregators to extract user interests from both positive and negative implicit feedback. The content-based aggregator focuses on the contents in the news representations and the denoising aggregator aims to mitigate the noise impact commonly existing in the implicit feedback. Besides, apart from the title information, DRPN also exploits the collaborative information by the graph neural network to further improve the recommendation performance. Experimental results on a large-scale public dataset demonstrate the state-of-the-art performance of DRPN. The further study results also show the effectiveness of the denoising module.

## Acknowledgements

This work was supported by National Key R&D Program of China, No. 2018YFC0117000.

## References

- 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 (1)*, pages 336–345. Association for Computational Linguistics.
- Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.
- Zhi Bian, Shaojun Zhou, Hao Fu, Qihong Yang, Zhenqi Sun, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2021. Denoising user-aware memory network for recommendation. In *RecSys*, pages 400–410. ACM.
- Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280.
- Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. 2012. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 153–162.
- Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph enhanced representation learning for news recommendation. In *Proceedings of The Web Conference 2020*, pages 2863–2869.
- Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: A factorization-machine based neural network for CTR prediction. In *IJCAI*, pages 1725–1731. ijcai.org.
- Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2020a. Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management*, 57(2):102142.
- Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020b. Graph neural news recommendation with unsupervised preference disentanglement. In *ACL*, pages 4255–4264. Association for Computational Linguistics.
- Wouter IJntema, Frank Goossen, Flavius Frasinca, and Frederik Hogenboom. 2010. Ontology-based news recommendation. In *Proceedings of the 2010 EDBT/ICDT Workshops*, pages 1–6.
- Miaomiao Li and Licheng Wang. 2019. [A survey on personalized news recommendation technology](#). *IEEE Access*, 7:145861–145879.
- Rui Liu, Huilin Peng, Yong Chen, and Dell Zhang. 2020. Hypernews: Simultaneous news recommendation and active-time prediction via a double-task deep neural network. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI)*, Yokohama, Japan.
- Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *KDD*, pages 1933–1942. ACM.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543. ACL.
- Zhaopeng Qiu, Yunfan Hu, and Xian Wu. 2022. [Graph neural news recommendation with user existing and potential interest modeling](#). *ACM Trans. Knowl. Discov. Data*, 16(5).
- Zhaopeng Qiu, Xian Wu, and Wei Fan. 2020. Automatic distractor generation for multiple choice questions in standard tests. In *COLING*, pages 2096–2106. International Committee on Computational Linguistics.

- Steffen Rendle. 2012. Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22.
- Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. 2021. [Masked label prediction: Unified message passing model for semi-supervised classification](#). In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1548–1554. International Joint Conferences on Artificial Intelligence Organization.
- Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018a. DKN: deep knowledge-aware network for news recommendation. In *WWW*, pages 1835–1844. ACM.
- Menghan Wang, Mingming Gong, Xiaolin Zheng, and Kun Zhang. 2018b. Modeling dynamic missingness of implicit feedback for recommendation. *Advances in neural information processing systems*, 31:6669.
- Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 373–381.
- Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019a. Neural news recommendation with attentive multi-view learning. In *IJCAI*, pages 3863–3869. ijcai.org.
- Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019b. NPA: neural news recommendation with personalized attention. In *KDD*, pages 2576–2584. ACM.
- Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019c. Neural news recommendation with multi-head self-attention. In *EMNLP/IJCNLP (1)*, pages 6388–6393. Association for Computational Linguistics.
- Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020a. [User modeling with click preference and reading satisfaction for news recommendation](#). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3023–3029. International Joint Conferences on Artificial Intelligence Organization. Main track.
- Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020b. MIND: A large-scale dataset for news recommendation. In *ACL*, pages 3597–3606. Association for Computational Linguistics.
- Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020c. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.
- Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2020. Deep feedback network for recommendation. *Proceedings of IJCAI-PRICAI*.
- Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1040–1048.
- Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. [Dan: Deep attention neural network for news recommendation](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5973–5980.

## A Discussion

### A.1 Limitations

In this paper, to better learn the representations, our method refines the historical behaviors of the user by the denoising manner. There are still some potential directions to further improve our approach. First, since the user profile in the experimental dataset only contains the historical behaviors and has no basic information (e.g., gender and age), our current approach doesn't support these features but they are widely used in practice. After these features are ready, we can convert them to embeddings and fuse them with the semantic interest representations obtained by two interest encoders to better represent the user. Second, the news generally contains many forms of features except for the title (such as the cover image and author information) and our approach will explore how to involve more features to better represent the news.

### A.2 Potential Risks

Our approach is based on the collaborative filtering, which may lead to that all of recommended news are similar to what the user has seen. This is a common problem faced by the majority of recommender systems. The concentration of a large number of similar information may narrow users' perspective and result in an imbalance on the personal information structure (Li and Wang, 2019). Our method can combine with some rule/human-based strategies (such as popularity based recommendation) to improve the recommendation diversity to alleviate this problem.