FeedRec: News Feed Recommendation with Various User Feedbacks

Chuhan Wu[1], Fangzhao Wu[2*], Tao Qi[1], Qi Liu[3], Xuan Tian[4], Jie Li[5], Wei He[5],
Yongfeng Huang[1], and Xing Xie[2]

[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Microsoft Research Asia, Beijing 100080, China   [3]University of Science and Technology of China, Hefei 230027, China
[4]Beijing Forestry University, Beijing 100083, China   [5]Microsoft STCA, Beijing 100080, China
{wuchuhan15,wufangzhao,taoqi.qt}@gmail.com,qiliuql@ustc.edu.cn,tianxuan@bjfu.edu.cn,
{jieli1,hewe,xingx}@microsoft.com,yfhuang@tsinghua.edu.cn

ABSTRACT

Accurate user interest modeling is important for news recommendation. Most existing methods for news recommendation rely on implicit feedbacks like click for inferring user interests and model training. However, click behaviors usually contain heavy noise, and cannot help infer complicated user interest such as dislike. Besides, the feed recommendation models trained solely on click behaviors cannot optimize other objectives such as user engagement. In this paper, we present a news feed recommendation method that can exploit various kinds of user feedbacks to enhance both user interest modeling and model training. We propose a unified user modeling framework to incorporate various explicit and implicit user feedbacks to infer both positive and negative user interests. In addition, we propose a strong-to-weak attention network that uses the representations of stronger feedbacks to distill positive and negative user interests from implicit weak feedbacks for accurate user interest modeling. Besides, we propose a multi-feedback model training framework to learn an engagement-aware feed recommendation model. Extensive experiments on a real-world dataset show that our approach can effectively improve the model performance in terms of both news clicks and user engagement.

CCS CONCEPTS

• **Information systems** → **Recommender systems**;

KEYWORDS

News recommendation, News feed, User feedback

---

*Corresponding Author

(a) Implicit weak feedbacks (click/skip) and explicit feedbacks (share/dislike)   (b) Implicit strong feedbacks (finish/quick close)
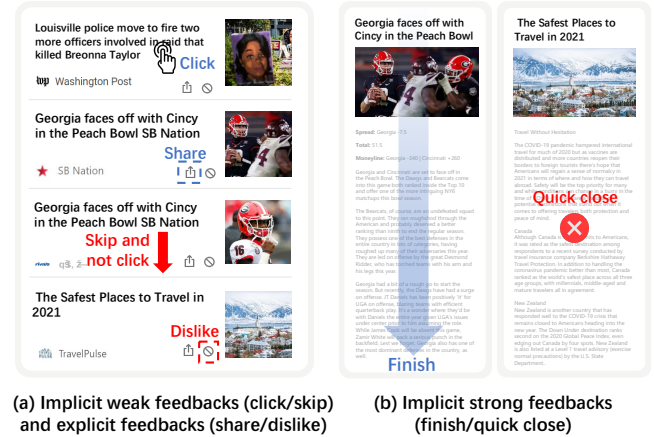
Figure 1: An example of various user feedbacks on a news feed platform.

1 INTRODUCTION

In recent years, online news feed services have gained huge popularity for users to obtain news information from never-ending feeds on their personal devices [14]. However, the huge volume of news articles streaming every day will overwhelm users [1]. Thus, personalized news recommendation is important for news feed services to alleviate information overload and improve the reading experience of users [11, 15, 43].

Most existing news recommendation methods rely on click behaviors of users to infer their interests and train the recommendation model [15, 27, 28, 30, 31, 34]. For example, Okura et al. [15] proposed to use a GRU network to learn user representations from historical clicked news. Wang et al. [28] proposed to use a candidate-aware attention network to measure the relevance between clicked news and candidate news when learning user representations. Wu et al. [34] proposed to use a combination of multi-head self-attention and additive attention networks to learn user representations from clicked news. All these methods are trained by predicting future news clicks based on the user interests inferred from historical clicked news. However, click behaviors are implicit feedbacks and usually contain heavy noise [29, 47]. For example, users may click a news due to the attraction of a news title but close it quickly if the user is disappointed at the news content [38]. In addition, many user interests such as like and dislike cannot be indicated by

the implicit click feedbacks, which are actually very important for improving the engagement of users on the news platform. Thus, it is insufficient to model user interests and train the recommendation model only based on news clicks.

Fortunately, on news feed platforms there are usually various kinds of user feedbacks. An example is shown in Fig. 1. Besides the weak implicit feedbacks such as click and skip, there are also explicit feedbacks like share and dislike (Fig. 1(a)) and strong implicit feedbacks like finishing the news article and closing the news webpage quickly after click (Fig. 1(b)). These feedbacks can provide more comprehensive information for inferring user interests [24]. However, it is non-trivial to incorporate the various feedbacks into news feed recommendation due to several challenges. First, implicit feedbacks are usually very noisy. Thus, it is important to distill real positive and negative user interests from the noisy implicit feedbacks. Second, different feedbacks have very different characteristics, e.g., the intensity of user interests they reflect. Thus, the model needs to take their differences into consideration. Third, the feedbacks of a user may have some inherent relatedness. For example, a user may quickly close the webpage of a clicked news and then push the dislike button. Thus, it is important to model the relatedness between feedbacks for better modeling user interests.

In this paper, we present a news feed recommendation approach named *FeedRec*[1] that can incorporate various user feedbacks into both user modeling and recommendation model training. In our method, we propose a unified framework to incorporate various explicit and implicit feedbacks of users, including *click*, *skip*, *share*, *dislike*, *finish*, and *quick close*, to infer both positive and negative interests of users.[2] We use a heterogeneous Transformer to capture the relatedness among different kinds of feedbacks, and use several homogeneous Transformers to capture the relations among the same kind of feedbacks. In addition, we propose a strong-to-weak attention network that uses the representations of stronger feedbacks to distill accurate positive and negative interests from implicit weak feedbacks. Besides, we propose a multi-feedback model training framework that jointly trains the model using click prediction, finish prediction and dwell time prediction tasks to learn an engagement-aware feed recommendation model. Extensive experiments on real-world dataset validate that our approach can not only gain more news clicks but also effectively improve user engagement in different aspects.

The contributions of this paper are summarized as follows:

- We propose a unified user modeling framework which can incorporate various explicit and implicit feedbacks to infer both positive and negative user interests.
- We propose a strong-to-weak attention network to distill accurate positive and negative user interests from implicit feedbacks with the guidance of strong feedbacks.
- We propose a multi-feedback model training framework by jointly training the model in click, finish and dwell time prediction tasks to learn engagement-aware feed recommendation models.

---

[1]Source code is available at https://github.com/wuch15/FeedRec.
[2]Our approach is a general framework to incorporate various user feedbacks and it is compatible with other types of feedbacks.

2 RELATED WORK

User modeling is critical for personalized news recommendation [10]. Most existing news recommendation approaches model user interests based on historical clicked news [2, 4–6, 8, 12, 16–21, 25, 30, 32, 36, 37, 39–42, 46, 48–50]. For example, Okura et al. [15] proposed an embedding-based news recommendation method that uses a GRU network to capture user interests from the representations of clicked news. Wang et al. [28] proposed to use a candidate-aware attention network to learn user representations from clicked news based on their relevance to candidate news. Wu et al. [31] proposed a news recommendation method with personalized attention network that selects informative clicked news for user modeling according to the embeddings of user IDs. Wu et al. [34] proposed to use multi-head self-attention mechanism to capture the relations between clicked news and use additive attention to select informative news for user modeling. Wang et al. [27] proposed to use a hierarchical dilated convolution neural network to learn multi-grained features of clicked news for representing users. These methods only consider the click behaviors of users. However, click behaviors are usually very noisy for inferring user interests because users may not click news only due to their interests. In addition, click behaviors cannot reflect many other kinds of user interests such as like or dislike. Thus, it is insufficient to accurately and comprehensively model user interests with click feedbacks only.

There are only a few news recommendation methods that consider user feedbacks beyond clicks in user modeling [13, 22, 33, 38, 44]. For example, Gershman et al. [3] proposed to represent users by the news they carefully read, rejected, and scrolled. Yi et al. [47] proposed to use the dwell time of news reading as the weights of clicked news for user modeling. Wu et al. [38] proposed a user modeling method based on click preference and reading satisfaction, which uses news clicks and the reading satisfaction derived from dwell time and news content length to model users. Xie et al. [44] proposed to model users' interests by their click, non-click and dislike feedbacks. They used click- and dislike-based user representations to distill positive and negative user interests from non-clicks, respectively. However, these methods mainly rely on clicked news to model the positive interests of users, which may not be accurate enough due to the heavy noise in click behaviors. Different from them, our approach can incorporate the various feedbacks of users into user modeling to distill both positive and negative feedbacks, which can capture user interests more comprehensively and accurately. In addition, our approach jointly trains the model in various tasks including click prediction, finish prediction and dwell time prediction, which can learn an engagement-aware feed recommendation model.

3 METHODOLOGY

In this section, we introduce the details of our *FeedRec* approach for news feed recommendation. We first introduce its user modeling framework, then describe the model architecture for news modeling, and finally introduce our multi-feedback model training method.

3.1 User Modeling

The user modeling framework of our *FeedRec* approach is shown in Fig. 2. It aims to accurately infer the user preferences for subsequent
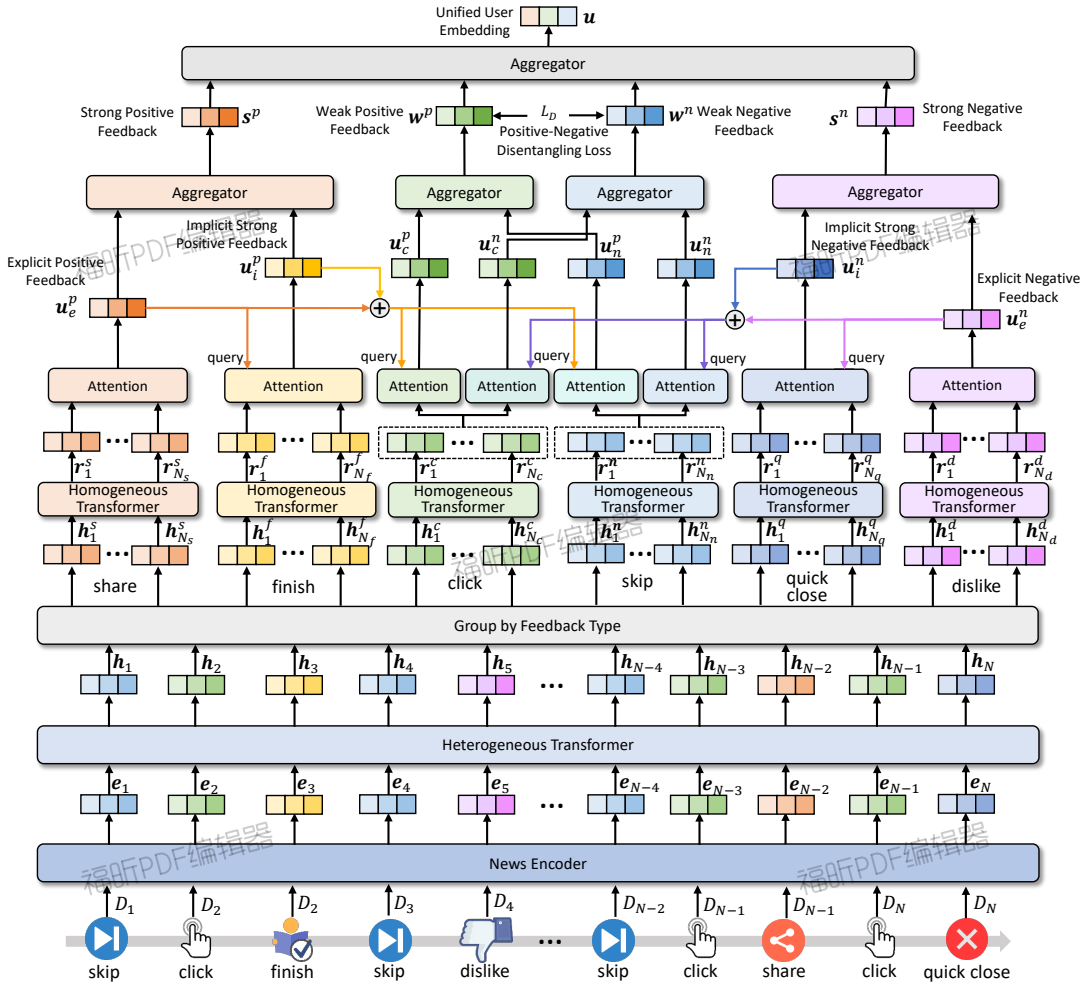
Figure 2: The user modeling framework of our *FeedRec* approach.

news feed recommendation by distilling positive and negative user interests from both explicit and implicit feedbacks it incorporates. In our approach, we consider six kinds of user feedbacks in total, including *click*, *skip*, *share*, *dislike*, *finish* and *quick close*. As shown in Fig. 1(a), the *click* feedback is obtained from users' click behaviors on the displayed news articles, which is a commonly used implicit positive feedback for user modeling. Users can also skip some news without click such as the third news in Fig. 1(a), which is regarded as an implicit negative feedback. In addition, along with each displayed news, there are buttons for users to provide explicit feedbacks such as *share* and *dislike*. For example, the user shares the second news in Fig. 1(a) while reports a dislike of the fourth news. Besides, there are also implicit feedback stronger than *click* and *skip*. For example, as shown in Fig. 1(b), after a user clicking a news, this user may finish reading this news (including watching the embedded video), which usually indicates a positive interest. However, the user may also take a quick read after click for only a few seconds and then close the news webpage, which is an indication of dissatisfaction.

We use the news reading behavior with dwell time shorter than T seconds to construct this kind of feedback.

Next, we introduce the architecture of our user modeling framework. We first use a shared news encoder to obtain the embedding of each feedback and its associated news article. We denote the feedback sequence as $[D_1, D_2, ..., D_N]$, where N is the sequence length.[3] It is converted into a feedback embedding sequence, which is denoted as $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N]$.

Next, we apply a heterogeneous feedback Transformer [26] to the feedback embedding sequence to capture the relations between different feedbacks. The feedbacks from the same user may have some inherent relatedness [44]. For example, the *finish* and *quick close* feedbacks usually appear after clicks. In addition, some skips may also have correlations to the previous clicks because a user may only choose to read a few news on similar topics [9]. For example, in Fig. 1(a) the user clicks and shares the second news while skips the third news, which may be because both of them are about the

---

[3]Some feedbacks may occur on the same news, e.g., finishing after clicking.

same football team. Thus, we use a heterogeneous feedback Transformer to capture the relations among various kinds of feedbacks in a feedback sequence. It receives the feedback embedding sequence $\mathbf{E}$ as the input, and outputs a hidden feedback representation sequence $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N]$. To help the subsequent user modeling process that separately models different kinds of feedbacks, we group the hidden feedback representations by their types. We denote the embedding sequences of *share, finish, click, skip, quick close* and *dislike* feedbacks respectively as $\mathbf{H}^s = [\mathbf{h}_1^s, \mathbf{h}_2^s, ...\mathbf{h}_{N_s}^s]$, $\mathbf{H}^f = [\mathbf{h}_1^f, \mathbf{h}_2^f, ...\mathbf{h}_{N_f}^f]$, $\mathbf{H}^c = [\mathbf{h}_1^c, \mathbf{h}_2^c, ...\mathbf{h}_{N_c}^c]$, $\mathbf{H}^n = [\mathbf{h}_1^n, \mathbf{h}_2^n, ...\mathbf{h}_{N_n}^n]$, $\mathbf{H}^q = [\mathbf{h}_1^q, \mathbf{h}_2^q, ...\mathbf{h}_{N_q}^q]$ and $\mathbf{H}^d = [\mathbf{h}_1^d, \mathbf{h}_2^d, ...\mathbf{h}_{N_d}^d]$, where N_s, N_f, N_c, N_n, N_q and N_d are the numbers of the corresponding feedbacks.

Following is a homogeneous feedback Transformer, which is applied to each kind of feedbacks to learn feedback-specific representations. Different kinds of feedbacks usually have very different characteristics. For example, *click* and *skip* feedbacks are usually abundant but noisy, while *share* and *dislike* feedbacks are strong but sparse. Thus, they may need to be handled differently. In addition, the relations between the same kind of feedbacks are also important for user interest modeling [44]. For example, researchers have found that modeling the interactions between clicked news can help better infer user interests [34]. Since the heterogeneous Transformer may not focus on capturing the relatedness between homogeneous feedback, we apply independent Transformers to each kind of feedbacks to learn feedback-specific representations for them and meanwhile capture the relations among homogeneous feedbacks. We denote the feedback-specific representation sequences of *share, finish, click, skip, quick close* and *dislike* as $\mathbf{R}^s = [\mathbf{r}_1^s, \mathbf{r}_2^s, ...\mathbf{r}_{N_s}^s]$, $\mathbf{R}^f = [\mathbf{r}_1^f, \mathbf{r}_2^f, ...\mathbf{r}_{N_f}^f]$, $\mathbf{R}^c = [\mathbf{r}_1^c, \mathbf{r}_2^c, ...\mathbf{r}_{N_c}^c]$, $\mathbf{R}^n = [\mathbf{r}_1^n, \mathbf{r}_2^n, ...\mathbf{r}_{N_n}^n]$, $\mathbf{R}^q = [\mathbf{r}_1^q, \mathbf{r}_2^q, ...\mathbf{r}_{N_q}^q]$ and $\mathbf{R}^d = [\mathbf{r}_1^d, \mathbf{r}_2^d, ...\mathbf{r}_{N_d}^d]$, respectively.

Based on the representation sequences of each kind of feedbacks, we then propose a strong-to-weak attention network to distill accurate positive and negative interests from implicit weak feedbacks (e.g., *clicks*) based on their relevance to stronger feedbacks (e.g., *share* and *finish*). Since explicit feedbacks like *share* and *dislike* are usually reliable, we can directly regard them as pure positive and negative feedbacks, respectively. We apply two separate attention networks [45] to them to learn an explicit positive feedback representation $\mathbf{u}_e^p$ and an explicit negative feedback representation $\mathbf{u}_e^n$, which are formulated as follows:

$$\alpha_k^p = \frac{\exp(\mathbf{q}^s \cdot \mathbf{r}_k^s)}{\sum_{j=1}^{N_s} \exp(\mathbf{q}^s \cdot \mathbf{r}_j^s)}, \quad \mathbf{u}_e^p = \sum_{k=1}^{N_s} \alpha_k^p \mathbf{r}_k^s, \tag{1}$$

$$\alpha_k^n = \frac{\exp(\mathbf{q}^d \cdot \mathbf{r}_k^d)}{\sum_{j=1}^{N_d} \exp(\mathbf{q}^d \cdot \mathbf{r}_j^d)}, \quad \mathbf{u}_e^n = \sum_{k=1}^{N_d} \alpha_k^n \mathbf{r}_k^d. \tag{2}$$

Next, we use the explicit positive feedback representation $\mathbf{u}_e^p$ to select informative *finish* feedbacks and build a representation $\mathbf{u}_i^p$ of implicit strong positive feedback, which is formulated as follows:

$$\beta_k^p = \frac{\exp(\mathbf{u}_e^p \cdot \mathbf{r}_k^f)}{\sum_{j=1}^{N_f} \exp(\mathbf{u}_e^p \cdot \mathbf{r}_j^f)}, \quad \mathbf{u}_i^p = \sum_{k=1}^{N_f} \beta_k^p \mathbf{r}_k^f. \tag{3}$$

The implicit strong negative feedback $\mathbf{u}_i^n$ is computed in a similar way from the representations of *quick close* feedbacks as follows:

$$\beta_k^n = \frac{\exp(\mathbf{u}_e^n \cdot \mathbf{r}_k^q)}{\sum_{j=1}^{N_q} \exp(\mathbf{u}_e^n \cdot \mathbf{r}_j^q)}, \quad \mathbf{u}_i^n = \sum_{k=1}^{N_q} \beta_k^n \mathbf{r}_k^q. \tag{4}$$

Click and *skip* feedbacks are usually noisy for inferring positive and negative interests [38, 44]. This is because clicks do not necessarily mean like or satisfaction, and those seen but skipped news may also be relevant to user interests. Thus, we need to distill the real positive and negative user interests from them. To address this problem, we select *click* and *skip* feedbacks based on their relevance to strong feedbacks for learning positive and negative user interest representations. We use the summation of $\mathbf{u}_e^p$ and $\mathbf{u}_i^p$ as the attention query for distilling the click-based and skip-based weak positive interests (denoted as $\mathbf{u}_c^p$ and $\mathbf{u}_n^p$), which are computed as follows:

$$\gamma_k^p = \frac{\exp[(\mathbf{u}_e^p + \mathbf{u}_i^p) \cdot \mathbf{r}_k^c]}{\sum_{j=1}^{N_c} \exp[\mathbf{u}_e^p + \mathbf{u}_i^p) \cdot \mathbf{r}_j^c]}, \quad \mathbf{u}_c^p = \sum_{k=1}^{N_c} \gamma_k^p \mathbf{r}_k^c, \tag{5}$$

$$\gamma_k^n = \frac{\exp[(\mathbf{u}_e^p + \mathbf{u}_i^p) \cdot \mathbf{r}_k^n]}{\sum_{j=1}^{N_n} \exp[\mathbf{u}_e^p + \mathbf{u}_i^p) \cdot \mathbf{r}_j^n]}, \quad \mathbf{u}_n^p = \sum_{k=1}^{N_n} \gamma_k^n \mathbf{r}_k^n. \tag{6}$$

The click- and skip-based weak negative feedbacks (denoted as $\mathbf{u}_c^n$ and $\mathbf{u}_n^n$) are computed similarly by using $\mathbf{u}_e^n + \mathbf{u}_i^n$ as the attention query. In this way, we can distill accurate positive and negative user interest information from the noisy feedbacks.

The last one is feedback aggregation. It aims to aggregate different kinds of feedbacks into summarized representations by considering their different importance and functions. We first aggregate the explicit positive feedback $\mathbf{u}_e^p$ and implicit strong positive feedback $\mathbf{u}_i^p$ into a unified strong positive feedback representation $\mathbf{s}^p$, which is formulated as follows:

$$\delta^p = \sigma(\mathbf{v}^p \cdot [\mathbf{u}_e^p; \mathbf{u}_i^p]), \mathbf{s}^p = \delta^p \mathbf{u}_e^p + (1 - \delta^p)\mathbf{u}_i^p, \tag{7}$$

where σ is the sigmoid function, $\mathbf{v}^p$ is a learnable vector. In a similar way, we aggregate the explicit negative feedback $\mathbf{u}_e^n$ and implicit strong negative feedback $\mathbf{u}_i^n$ into a unified strong negative feedback representation $\mathbf{u}^n$ as follows:

$$\delta^n = \sigma(\mathbf{v}^n \cdot [\mathbf{u}_e^n; \mathbf{u}_i^n]), \mathbf{s}^n = \delta^n \mathbf{u}_e^n + (1 - \delta^n)\mathbf{u}_i^n, \tag{8}$$

where $\mathbf{v}^n$ are parameters. Similarly, we aggregate the click-based and skip-based positive feedbacks ($\mathbf{u}_c^p$ and $\mathbf{u}_n^p$) into a weak positive feedback representation $\mathbf{w}^p$, and aggregate $\mathbf{u}_c^n$ and $\mathbf{u}_n^n$ into a weak negative feedback representation $\mathbf{w}^n$. We finally aggregate the four kinds of feedbacks, i.e., $\mathbf{s}^p$, $\mathbf{w}^p$, $\mathbf{w}^n$ and $\mathbf{s}^n$ into a unified user embedding $\mathbf{u}$, which is formulated as follows:

$$\mathbf{u} = s^p \mathbf{s}^p + w^p \mathbf{w}^p + s^n \mathbf{s}^n + w^n \mathbf{w}^n, \tag{9}$$

where s^p, w^p, s^n, w^n are learnable parameters.

3.2 News Modeling

In this section, we briefly introduce the details of news encoder in our approach. The architecture of the news encoder is shown in Fig. 3. For each feedback on news, we compute five kinds of embeddings for it. The first one is text embedding, which is computed from news title through a Transformer [26] network to capture the
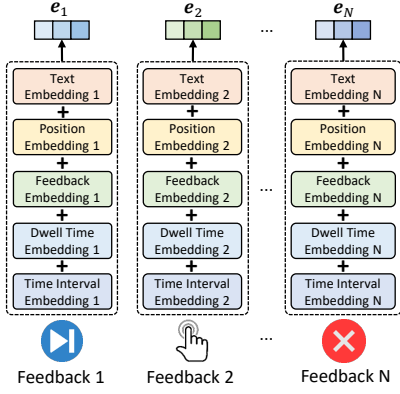
Figure 3: The architecture of the news encoder.

Figure 4: The multi-feedback model training framework.

semantic information of news. The second one is position embedding, which aims to encode the positional information of feedback. The third one is feedback embedding, which encodes the type of feedback to help better distinguish different kinds of feedbacks.[4] The fourth one is dwell time embedding, which aims to encode use engagement information (we use the user-specific dwell time). We use a quantization function $\tilde{t} = \lfloor \log_2(t+1) \rfloor$ to convert the real-valued dwell time t into a discrete value $\tilde{t}$ for building the embedding table. The last one is time interval embedding, which aims to better capture the relatedness between adjacent feedbacks. We use the same quantization function to convert the time interval between the current and previous feedbacks into a discrete variable for embedding. These embeddings are added together into a unified news embedding for subsequent user modeling and model training.

3.3 Multi-feedback Model Training

In this section, we introduce the multi-feedback framework in our approach. Existing news recommendation methods mainly rely on the click signals to train the recommendation model. However, there are usually some gaps between news clicks and user engagement or satisfaction, because users may leave the news page quickly if they are not satisfied with the quality of news content. Thus, we propose to jointly train the model in three tasks, including click prediction, finish prediction and dwell time prediction, to encode both click and user engagement information. The model training framework is shown in Fig. 4. We use the user encoder to learn a user embedding $\mathbf{u}$ from the feedback sequence and use the news encoder to encode the candidate news into its embedding $\mathbf{e}$. We denote the predicted click, finish and dwell time scores of this pair of user and candidate news as $\hat{y}$, $\hat{z}$ and $\hat{t}$ respectively, which are computed as follows:

$$\hat{y} = \mathbf{u} \cdot \mathbf{e},$$
$$\hat{z} = \mathbf{u} \cdot (\mathbf{W}_z \mathbf{e}), \quad (10)$$
$$\hat{t} = \max[0, \mathbf{u} \cdot (\mathbf{W}_t \mathbf{e})],$$

where $\mathbf{W}_z$ and $\mathbf{W}_t$ are learnable parameters.

Following [34], we use negative sampling techniques to construct training samples. For each clicked news, we sample K skipped news

---
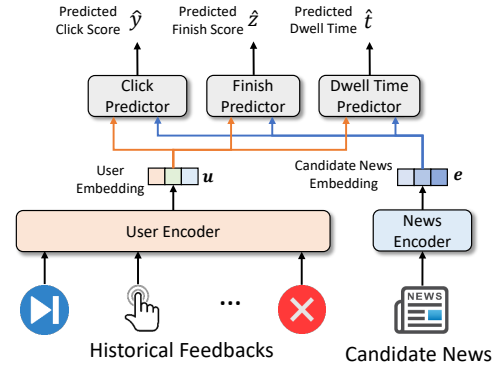[4]This embedding is deactivated when encoding candidate news.

displayed on the same page, and jointly predict the three kinds of scores for these $K+1$ news. The click, finish and dwell time prediction losses on a sample are formulated as follows:

$$\mathcal{L}_R = -\log\left[\frac{\exp(\hat{y}^+)}{\exp(\hat{y}^+) + \sum_{i=1}^{K} \exp(\hat{y}_i^-)}\right],$$
$$\mathcal{L}_F = -z^+ \log[\sigma(\hat{z}^+)] - (1 - z^+) \log[1 - \sigma(\hat{z}^+)], \quad (11)$$
$$\mathcal{L}_T = |t^+ - \hat{t}^+|,$$

where $\hat{y}^+$ and $\hat{y}_i^-$ are the predicted click scores for a clicked news and its associated i-th skipped news. $\hat{z}^+$, z^+, $\hat{t}_i^+$ and t_i^+ stand for the predicted finish label, real finish label, predicted dwell time, and real dwell time of a clicked news, respectively.[5]

Besides, since we expect the weak positive feedback to be different from the weak negative feedback, we propose a positive-negative disentangling loss $\mathcal{L}_d$ to help distill more accurate positive and negative user interests by regularizing $\mathbf{w}^p$ and $\mathbf{w}^n$ as follows:

$$\mathcal{L}_D = \frac{\mathbf{w}^p \cdot \mathbf{w}^n}{||\mathbf{w}^p|| \times ||\mathbf{w}^n||}, \quad (12)$$

where $||\cdot||$ means the L2-norm. The final unified loss $\mathcal{L}$ is a weighted summation of four loss functions, which is formulated as follows:

$$\mathcal{L} = \mathcal{L}_R + \alpha \mathcal{L}_F + \beta \mathcal{L}_T + \gamma \mathcal{L}_D, \quad (13)$$

where α, β and γ are loss coefficients that control the relative importance of the corresponding loss functions.

4 EXPERIMENTS

4.1 Dataset and Experimental Settings

In our experiments, since there is no off-the-shelf dataset for news recommendation that contains multiple kinds of user feedbacks, we constructed one by ourselves from a commercial news feed App. The dataset contains the behavior logs of 10,000 users in about one month, i.e., from Sep. 1st, 2020 to Oct. 2nd, 2020. The logs in the last week were used for test, and the rest ones were used for training and validation (rest logs on the last day). The statistics of this dataset is shown in Table 1. We can see that explicit feedbacks like *share* and *dislike* are relatively sparse, while implicit feedbacks are much richer. The distributions of the number of each kind of feedback provided by a user are shown in Fig. 5. We

---
[5]We use the log function to transform the raw dwell time and then normalize it.

Table 1: Detailed statistics of the datasets.

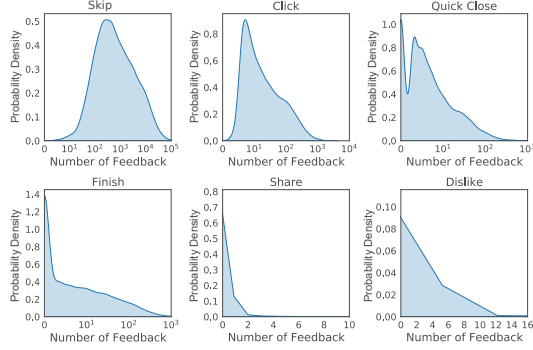| # user	10,000	# news	590,485
# impression	351,581	# click	493,266
# skip	25,986,877	# share	2,764
# dislike	17,073	# finish	234,759
# quick close	108,396	avg. dwell time	83.90s

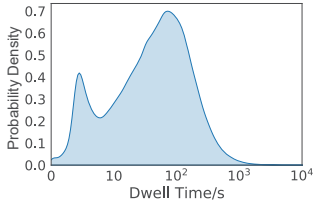**Figure 5: Distribution of different kinds of feedbacks.**

Figure 6: Dwell time distribution of the dataset.

can find that the number of *skip* feedbacks is approximately log-normal, while the numbers of other kinds of feedbacks obey long-tail distributions. Since *skip* feedbacks are dominant in our dataset, we only randomly sample 10% of skips to reduce the length of input sequence. We also show the distribution of dwell time in our dataset in Fig. 6. We find an interesting phenomenon is that the distribution has two peaks, one of which approximately appears between 0 and 10 seconds. This may be because users are sometimes disappointed at the news content and quickly close the webpage. Thus, we accordingly set the dwell time threshold T to 10 seconds to construct the *quick close* feedbacks, and we will discuss the influence of T in the hyperparameter analysis section.

In our experiments, we followed the same settings in [34] to generate the 256-dim text embeddings, and the dimensions of other embeddings in the news encoder were also 256. The Transformers in the user modeling part had 16 heads, and the output dimension of each head was 16. The feedback type, position, dwell time, and time interval embeddings are randomly initialized. The optimizer for model training was Adam [7], and the learning rate was 1e-4. The negative sampling ratio was 4. The batch size was 32. The dropout [23] ratio was set to 0.2. These hyperparameters were tuned on the validation sets. We used AUC, MRR, nDCG@5 and HR@5 to

Table 2: Performance comparison in terms of news clicks.

Methods	AUC	MRR	nDCG@5	HR@5
EBNR [15]	0.6112	0.2622	0.2790	0.1062
DKN [28]	0.6076	0.2591	0.2768	0.1045
NPA [31]	0.6210	0.2685	0.2882	0.1095
NAML [30]	0.6192	0.2670	0.2871	0.1089
LSTUR [1]	0.6224	0.2701	0.2896	0.1099
NRMS [34]	0.6231	0.2707	0.2904	0.1103
FIM [27]	0.6250	0.2729	0.2925	0.1114
DFN [44]	0.6296	0.2748	0.2948	0.1140
CPRS [38]	0.6334	0.2781	0.2972	0.1156
FeedRec	**0.6609**	**0.3026**	**0.3304**	**0.1328**

Table 3: Performance comparison in terms of user engagement. ↑ Means higher is better, while ↓ means lower is better.

Methods	Share(↑)	Dislike(↓)	Finish(↑)	Dwell Time/s(↑)
EBNR [15]	1.1203	0.9679	0.0671	84.061
DKN [28]	1.1169	0.9729	0.0655	83.494
NPA [31]	1.1288	0.9588	0.0691	84.579
NAML [30]	1.1269	0.9593	0.0689	84.487
LSTUR [1]	1.1325	0.9610	0.0696	84.712
NRMS [34]	1.1343	0.9583	0.0709	84.793
FIM [27]	1.1365	0.9595	0.0711	85.010
DFN [44]	1.1398	0.9519	0.0745	85.346
CPRS [38]	1.1455	0.9434	0.0772	86.129
FeedRec	**1.2603**	**0.9011**	**0.0940**	**87.989**

measure the click-based model performance. In addition, we used several metrics to measure the model performance in terms of user engagement. We used the ratio of the *share/dislike* frequency of top 5 ranked news to the overall *share/dislike* frequency in the dataset to measure *share/dislike* based performance, and we also reported the average finishing ratio of top 5 ranked news and their average dwell time if clicked. We independently repeated each experiment 5 times and reported the average results.

4.2 Performance Evaluation

First, we compare the performance of our *FeedRec* approach with many baseline methods, including: (1) EBNR [15], an embedding-based news recommendation method with GRU network; (2) DKN [28], deep knowledge network for news recommendation; (3) NPA [31], a neural news recommendation method with personalized attention; (4) NAML [30], a neural news recommendation method with attentive multi-view learning; (5) LSTUR [1], a news recommendation method that models long- and short-term user interests; (6) NRMS [34], using multi-head self-attention for news and user modeling; (7) FIM [27], a fine-grained interest matching approach for news recommendation; (8) DFN [44], deep feedback network for feed recommendation; (9) CPRS [38], a news recommendation approach with click preference and reading satisfaction. The click-based and user-engagement performance of different methods are shown in Tables 2 and 3, respectively. We have several findings from the results. First, compared with the methods based on click feedbacks only, the methods that consider other user feedbacks (i.e.,
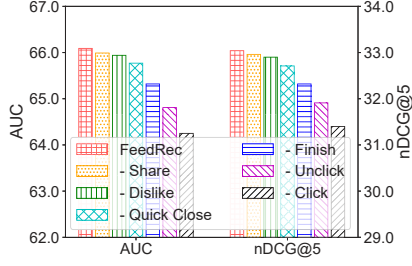
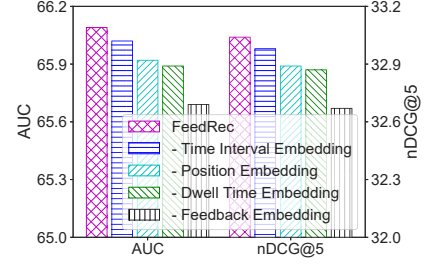Figure 7: Influence of different types of user feedbacks.

Figure 8: Effectiveness of several core model components.
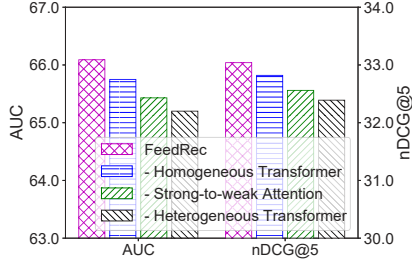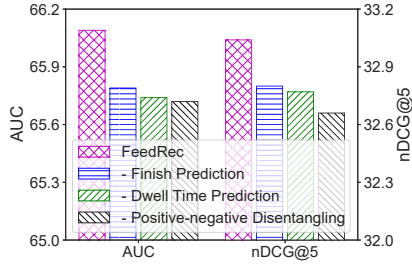

Figure 9: Influence of different loss functions.

DFN, *CPRS* and *FeedRec*) achieve better performance in terms of news clicks and user engagement. It shows that *click* feedbacks may not be sufficient to model user interests accurately and other feedbacks such as *dislike* and dwell time can provide complementary information for user modeling. Second, among the methods that can exploit multiple kinds of user feedbacks, *CPRS* and *FeedRec* perform better than *DFN*. This may be because the *dislike* feedbacks are relatively sparse, which may be insufficient to distill negative user interests accurately. Third, our *FeedRec* approach outperforms other compared methods in both click- and engagement-based metrics. This is probably because our approach can effectively exploit the various feedbacks of users to model their interests more accurately. In addition, our multi-feedback model training framework not only considers news clicks but also the engagement signals, which can help learn a user engagement-aware recommendation model to improve user experience.

Figure 10: Effect of different embeddings in news encoder.

Figure 11: Influence of the dwell time threshold T.

4.3 Influence of Different Feedbacks

Next, we study the influence of different feedbacks on the model performance. We compare the performance of *FeedRec* and its variants with one kind of feedbacks removed, and the results are shown in Fig. 7. We find that the performance declines when any kind of feedbacks is dropped. Among them, the *click* feedback plays the most important role, which is intuitive. However, we find it is interesting that the *skip* feedback is the second most important. This may be because skips can also provide rich clues for inferring user interests (usually negative ones) to support user modeling. In addition, *finish* and *quick close* feedbacks are also important. This may be because both kinds of feedbacks are indications of users' news reading satisfaction, which are important for modeling user preferences. Besides, *share* and *dislike* feedbacks are also useful, but their contributions are relatively small. This may be because that although these explicit feedbacks are strong indications of user preference, they are usually sparse in practice. Thus, it is important to incorporate other implicit feedbacks like finish to model user interests more comprehensively.

4.4 Model Effectiveness

Then, we validate the effectiveness of the core model components in our *FeedRec* approach and the loss functions used for model training. We first compare the performance of our approach and its variants with one component removed, as shown in Fig. 8. From the results, we find that the heterogeneous feedback Transformer contributes most. This may be because the heterogeneous feedback Transformer can capture the global relatedness between the feedbacks of a user. In addition, the strong-to-weak attention network is also very useful. This is because it can select informative feedbacks for user modeling and meanwhile take the information of strong

(a) Finish prediction loss coefficient α. (b) Dwell time prediction loss coefficient β. (c) Positive-negative disentangling loss γ.
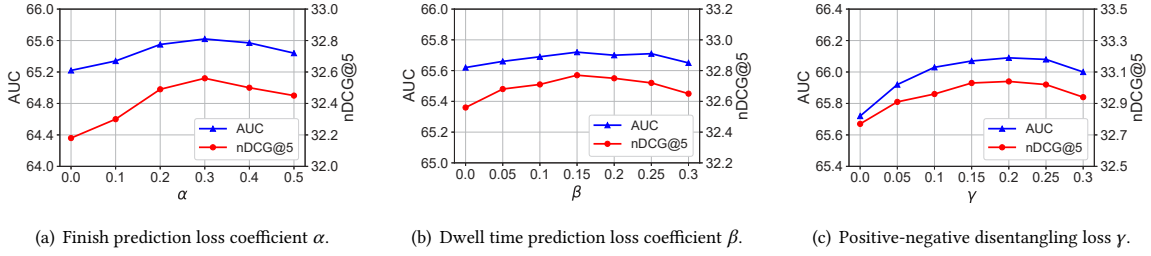
Figure 12: Influence of different loss coefficients on the model performance.

feedbacks into consideration, which can help distill positive and negative user interests more precisely. Moreover, the homogeneous Transformer can also improve the performance. This may be because it can better capture the diverse characteristics of different kinds of feedbacks and benefit user modeling.

We also study the influence of each loss function on model training by removing it from the unified training loss. The results are shown in Fig. 9. We find that the positive-negative disentangling loss can effectively improve the model performance. This may be because it can push the model to distill positive and negative interest information more accurately, which is beneficial for recommendation. In addition, both the finish prediction and dwell time prediction losses are helpful. This may be because finish and dwell time signals are correlated to user satisfaction. Thus, incorporating these signals into model training can help learn an engagement-aware user model to improve the recommendation performance.

Finally, we investigate the influence of several different kinds of embeddings in the news encoder, including position embedding, feedback embedding, dwell time embedding and time interval embedding by removing one of them.[6] We illustrate the results in Fig. 10. We find the feedback embedding plays the most important role. This is because the embedding of feedback type is very useful for distinguishing different kinds of feedbacks. In addition, the dwell time embedding is also important. This may be because dwell time embeddings can provide rich information on inferring the satisfaction of users. Besides, both position and time interval embeddings are useful. This is because position embeddings can help capture the feedback orders and time interval embeddings can help better model the relatedness between adjacent feedbacks.

4.5 Hyperparameter Analysis

In this section, we present some analysis on several critical hyperparameters in our approach, including the dwell time threshold T for constructing *quick close* feedbacks and the coefficients (i.e., α, β and γ) for controlling the importance of different tasks. We first vary the threshold T from 0 to 25 seconds to study its influence on model performance. The results are shown in Fig. 11. We find that the performance is suboptimal when the threshold T is too small (e.g., 5 seconds). This may be because many negative feedbacks with short reading dwell time cannot be exploited. In addition, the performance also declines when T goes too large. This is because

---

[6]We do not report the scores without text embeddings because the performance is quite unsatisfactory.

many positive feedbacks will be mistakenly regarded as negative ones, which is not beneficial for user interest modeling. Thus, in our approach the threshold T is set to 10 seconds, which is also consistent with the findings in [35].

We then study the influence of the three loss coefficients. We first tune the finish prediction loss coefficient α under $\beta = \gamma = 0$. The results are shown in Fig. 12(a). We find that the performance is not optimal when α is either too small or too large. This may be because the finish signals are not fully exploited when α is very small, while the main click prediction task will be influenced if the coefficient goes too large. Thus, we empirically set α to 0.2. Then, we tune the dwell time prediction loss coefficient β under $\alpha = 0.2$ and $\gamma = 0$. The results are shown in Fig. 12(b). We find that there is also a peak on the performance curve. This may be because the dwell time signals cannot be effectively captured if β is too small, while the click prediction task is not fully respected when β is too large. Thus, we set β to 0.15 according to the results. Finally, we search the value of the positive-negative disentangling loss coefficient γ under the previous settings of α and β. We observe that a moderate value of γ such as 0.2 is suitable for our approach. This may be because the positive and negative feedbacks cannot be effectively distinguished when γ is too small, while this regularization loss is over emphasized when γ is too large.

5 CONCLUSION

In this paper, we present a general news feed recommendation approach that can exploit various kinds of user feedbacks with different intensities. In our approach, we propose a unified user modeling framework to incorporate various explicit and implicit user feedbacks to comprehensively capture user interests. In addition, we propose a strong-to-weak attention network that uses strong feedbacks to distill accurate positive and negative user interests from weak implicit feedbacks. Besides, we propose a multi-feedback model training framework to train the model in the click, finish and dwell time prediction tasks to learn engagement-aware feed recommendation models. Extensive experiments on real-world dataset validate that our approach can effectively improve model performance in terms of both news clicks and user engagement.

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*. 336–345.

[2] Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph Enhanced Representation Learning for News Recommendation. In *WWW*. 2863–2869.

[3] Anatole Gershman, Travis Wolfe, Eugene Fink, and Jaime G Carbonell. 2011. News personalization using support vector machines. (2011).

[4] Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2020. Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management* 57, 2 (2020), 102142.

[5] Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph neural news recommendation with unsupervised preference disentanglement. In *ACL*. 4255–4264.

[6] Dhruv Khattar, Vaibhav Kumar, Vasudeva Varma, and Manish Gupta. 2018. Weave& rec: A word embedding based 3-d convolutional network for news recommendation. In *CIKM*. ACM, 1855–1858.

[7] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[8] Dongho Lee, Byungkook Oh, Seungmin Seo, and Kyong-Ho Lee. 2020. News Recommendation with Topic-Enriched Knowledge Graphs. In *CIKM*. 695–704.

[9] Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. 2011. SCENE: a scalable two-stage personalized news recommendation system. In *SIGIR*. 125–134.

[10] Miaomiao Li and Licheng Wang. 2019. A Survey on Personalized News Recommendation Technology. *IEEE Access* 7 (2019), 145861–145879.

[11] Yuchen Li, Dongxiang Zhang, Ziquan Lan, and Kian-Lee Tan. 2016. Context-aware advertisement recommendation for high-speed social news feeding. In *ICDE*. IEEE, 505–516.

[12] Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. 2020. KRED: Knowledge-Aware Document Representation for News Recommendations. In *Recsys*. 200–209.

[13] Mingyuan Ma, Sen Na, Hongyu Wang, Congzhou Chen, and Jin Xu. 2021. The graph-based behavior-aware recommendation for interactive news. *Applied Intelligence* (2021), 1–17.

[14] Nuno Moniz and Luís Torgo. 2018. Multi-source social feedback of online news feeds. *arXiv preprint arXiv:1801.07055* (2018).

[15] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *KDD*. 1933–1942.

[16] Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021. Personalized News Recommendation with Knowledge-aware Interactive Matching. In *SIGIR*. 61–70.

[17] Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021. PP-Rec: News Recommendation with Personalized User Interest and Time-aware News Popularity. In *ACL*. 5457–5467.

[18] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-Preserving News Recommendation Model Learning. In *EMNLP: Findings*. 1423–1432.

[19] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2021. Uni-FedRec: A Unified Privacy-Preserving News Recommendation Framework for Model Training and Online Serving. In *EMNLP: Findings*. 1438–1448.

[20] Tao Qi, Fangzhao Wu, Chuhan Wu, Peiru Yang, Yang Yu, Xing Xie, and Yongfeng Huang. 2021. HieRec: Hierarchical User Interest Modeling for Personalized News Recommendation. In *ACL*.

[21] TYSS Santosh, Avirup Saha, and Niloy Ganguly. 2020. MVL: Multi-View Learning for News Recommendation. In *SIGIR*. 1873–1876.

[22] Shaoyun Shi, Weizhi Ma, Zhen Wang, Min Zhang, Kun Fang, Jingfang Xu, Yiqun Liu, and Shaoping Ma. 2021. WG4Rec: Modeling Textual Content with Word Graph for News Recommendation. In *CIKM*. 1651–1660.

[23] 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.

[24] Liang Tang, Bo Long, Bee-Chung Chen, and Deepak Agarwal. 2016. An empirical study on recommendation with multiple types of feedback. In *KDD*. 283–292.

[25] Yu Tian, Yuhao Yang, Xudong Ren, Pengfei Wang, Fangzhao Wu, Qian Wang, and Chenliang Li. 2021. Joint Knowledge Pruning and Recurrent Graph Convolution for News Recommendation. In *SIGIR*. 51–60.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.

[27] Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained Interest Matching for Neural News Recommendation. In *ACL*. 836–845.

[28] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *WWW*. 1835–1844.

[29] Hongyi Wen, Longqi Yang, and Deborah Estrin. 2019. Leveraging post-click feedback for content recommendations. In *Recsys*. 278–286.

[30] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Attentive Multi-View Learning. In *IJCAI*. 3863–3869.

[31] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Npa: Neural news recommendation with personalized attention. In *KDD*. 2576–2584.

[32] Chuhan Wu, Fangzhao Wu, Mingxiao An, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Topic-Aware News Representation. In *ACL*. 1154–1159.

[33] Chuhan Wu, Fangzhao Wu, Mingxiao An, Tao Qi, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with heterogeneous user behavior. In *EMNLP-IJCNLP*. 4874–4883.

[34] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Multi-Head Self-Attention. In *EMNLP-IJCNLP*. 6390–6395.

[35] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2020. Neural news recommendation with negative feedback. *CCF Transactions on Pervasive Computing and Interaction* 2, 3 (2020), 178–188.

[36] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2021. User-as-Graph: User Modeling with Heterogeneous Graph Pooling for News Recommendation. In *IJCAI*.

[37] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. SentiRec: Sentiment Diversity-aware Neural News Recommendation. In *AACL*. 44–53.

[38] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. User Modeling with Click Preference and Reading Satisfaction for News Recommendation. In *IJCAI-PRICAI*. 3023–3029.

[39] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering News Recommendation with Pre-trained Language Models. In *SIGIR*. 1652–1656.

[40] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Two Birds with One Stone: Unified Model Learning for Both Recall and Ranking in News Recommendation. *arXiv preprint arXiv:2104.07404* (2021).

[41] Chuhan Wu, Fangzhao Wu, Xiting Wang, Yongfeng Huang, and Xing Xie. 2021. FairRec:Fairness-aware News Recommendation with Decomposed Adversarial Learning. In *AAAI*. 4462–4469.

[42] Chuhan Wu, Fangzhao Wu, Yang Yu, Tao Qi, Yongfeng Huang, and Qi Liu. 2021. NewsBERT: Distilling Pre-trained Language Model for Intelligent News Application. In *EMNLP: Findings*. 3285–3295.

[43] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. MIND: A Large-scale Dataset for News Recommendation. In *ACL*. 3597–3606.

[44] Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2020. Deep Feedback Network for Recommendation. In *IJCAI-PRICAI*. 2519–2525.

[45] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL-HLT*. 1480–1489.

[46] Jingwei Yi, Fangzhao Wu, Chuhan Wu, Ruixuan Liu, Guangzhong Sun, and Xing Xie. 2021. Efficient-FedRec: Efficient Federated Learning Framework for Privacy-Preserving News Recommendation. In *EMNLP*. 2814–2824.

[47] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. 2014. Beyond clicks: dwell time for personalization. In *RecSys*. 113–120.

[48] Hui Zhang, Xu Chen, and Shuai Ma. 2019. Dynamic News Recommendation with Hierarchical Attention Network. In *ICDM*. IEEE, 1456–1461.

[49] Qi Zhang, Qinglin Jia, Chuyuan Wang, Jingjie Li, Zhaowei Wang, and Xiuqiang He. 2021. AMM: Attentive Multi-field Matching for News Recommendation. In *SIGIR*. 1588–1592.

[50] Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. Dan: Deep attention neural network for news recommendation. In *AAAI*, Vol. 33. 5973–5980.