

# Two Birds with One Stone: Unified Model Learning for Both Recall and Ranking in News Recommendation

Chuhan Wu<sup>†</sup> Fangzhao Wu<sup>‡\*</sup> Tao Qi<sup>†</sup> Yongfeng Huang<sup>†</sup>

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

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

{wuchuhan15, wufangzhao, taoqi.qt}@gmail.com  
yfhuang@tsinghua.edu.cn

## Abstract

Recall and ranking are two critical steps in personalized news recommendation. Most existing news recommender systems conduct personalized news recall and ranking separately with different models. However, maintaining multiple models leads to high computational cost and poses great challenges to meeting the online latency requirement of news recommender systems. In order to handle this problem, in this paper we propose UniRec, a unified method for recall and ranking in news recommendation. In our method, we first infer user embedding for ranking from the historical news click behaviors of a user using a user encoder model. Then we derive the user embedding for recall from the obtained user embedding for ranking by using it as the attention query to select a set of basis user embeddings which encode different general user interests and synthesize them into a user embedding for recall. The extensive experiments on benchmark dataset demonstrate that our method can improve both efficiency and effectiveness for recall and ranking in news recommendation.

## 1 Introduction

News recommendation techniques are widely used by many online news websites and Apps to provide personalized news services (Wu et al., 2020b). Recall and ranking are two critical steps in personalized news recommender systems (Karimi et al., 2018; Wu et al., 2021a). As shown in Fig. 1, when a user visits a news platform, the recommender system first recalls a set of candidate news from a large-scale news pool, and then ranks candidate news for personalized news display (Wu et al., 2020b). Both news recall and ranking have been widely studied (Elkahky et al., 2015; Liu et al., 2019, 2020; Wu et al., 2020a; Wang et al., 2020; Wu et al., 2021c; Qi et al., 2021a,b,c,d). In online news recommender systems, recall and ranking are

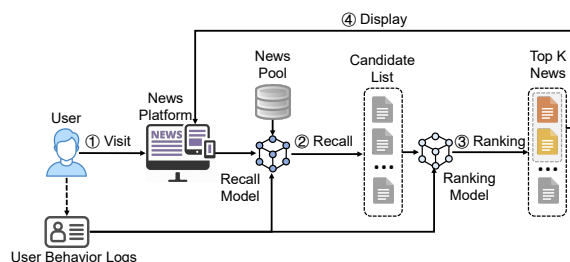


Figure 1: A typical pipeline of news recommendation.

usually conducted separately with different models, as shown in Fig. 1. However, maintaining separate models for news recall and ranking in large-scale news recommender systems usually leads to heavy computation and memory cost (Tan et al., 2020), and it may be difficult to meet the latency requirement of online news services.

Learning a unified model for personalized news recall and ranking would be greatly beneficial for alleviating the computation load of news recommender systems. However, it is a non-trivial task because the goals of recall and ranking are not the same (Covington et al., 2016; Malkov and Yashunin, 2018). Ranking usually aims to accurately rank candidates based on their relevance to user interests (Wu et al., 2019b; Ge et al., 2020; Wu et al., 2021b; Wang et al., 2020), while recall mainly aims to form a candidate pool that can comprehensively cover user interests (Liu et al., 2020; Qi et al., 2021d). Thus, the model needs to adapt to the different goals of recall and ranking without hurting their performance.

In this paper, we propose a news recommendation method named UniRec, which can learn a unified user model for personalized news recall and ranking. In our method, we first encode news into embeddings with a news encoder, and learn a user embedding for ranking from the embeddings of historical clicked news. We further derive the user embedding for recall by using the user embedding for ranking as the attention query to select a

\*Corresponding author.

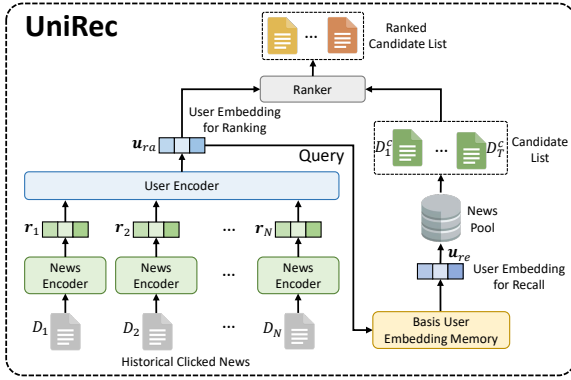


Figure 2: The framework of *UniRec*.

set of basis user embeddings that encode different general user interest aspects and synthesize them into a user embedding for recall. In the test phase, we only use the basis user embeddings with top attention weights to compose the user embedding for recall to filter noisy user interests. Extensive experiments on a real-world dataset demonstrate that our method can conduct personalized news recall and ranking with a unified model and meanwhile achieve promising recall and ranking performance.

## 2 Methodology

The overall framework of *UniRec* is shown in Fig. 2. We first learn a user embedding for ranking from the user’s historical clicked news. We then derive a user embedding for recall from the user embedding for ranking and a set of basis user embeddings that encode different general interests. Their details are introduced as follows.

### 2.1 Ranking for News Recommendation

The ranking part aims to rank candidate news in a small candidate list according to user interests. Following (Wu et al., 2020b), *UniRec* uses a news encoder that learns news embeddings from news texts and a user encoder that learns user interest embedding for ranking from the embeddings of clicked news. The candidate news embedding and user embedding for ranking are used to compute a click score for personalized news ranking. More specifically, we denote a user  $u$  has  $N$  historical clicked news  $[D_1, D_2, \dots, D_N]$ . These clicked news are encoded into a sequence of news embeddings, which is denoted as  $[r_1, r_2, \dots, r_N]$ . The user encoder further takes this sequence as input, and outputs a user embedding  $\mathbf{u}_{ra}$  for ranking. For a candidate news  $D_i^c$ , we use the news encoder to obtain its embedding  $\mathbf{r}_i^c$ . We follow (Okura et al., 2017) to

compute the probability score of the user  $u$  clicking on the candidate news  $D_i^c$  via inner product, i.e.,  $\hat{y}_{ra}^i = \mathbf{u}_{ra} \cdot \mathbf{r}_i^c$ . The click scores of the news in a candidate list are used for personalized ranking. Following (Wu et al., 2019c), we use multi-head self-attention networks in both news and user encoders to capture the contexts of words and click behaviors, respectively. In addition, following (Devlin et al., 2019) we add position embeddings to capture the orders of words and behaviors.

### 2.2 Recall for News Recommendation

The recall part aims to select candidate news from a large news pool based on their relevance to user interests. To efficiently exploit user interest information for personalized news recall, we take the user embedding for ranking as input instead of rebuilding user interest representations from original user click behaviors. However, since the goals of ranking and recall are not the same (Kang and McAuley, 2019), the user embedding for ranking may not be suitable for news recall. Thus, we propose a method to distill a user embedding for recall from the user embedding for ranking. More specifically, we maintain a basis user embedding memory that encodes different general user interest aspects. We denote the  $M$  basis user embeddings in the memory as  $[\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_M]$ . We use the user embedding for ranking as the attention query to select basis user embeddings. We denote the attention weight of the  $i$ -th basis user embedding as  $\alpha_i$ , which is computed as:

$$\alpha_i = \frac{\exp(\mathbf{u}_{ra} \cdot \mathbf{w}_i)}{\sum_{j=1}^M \exp(\mathbf{u}_{ra} \cdot \mathbf{w}_j)}, \quad (1)$$

where the parameters  $\mathbf{w}_i$  are served as the attention keys. Different from additive attention (Yang et al., 2016) where the attention keys and values are equivalent, in our approach the keys (i.e.,  $\mathbf{w}_i$ ) are different from the values (i.e.,  $\mathbf{v}_i$ ). This is because we expect the basis user embeddings to have different spaces with the user embeddings for ranking to better adapt to the recall task. The basis user embeddings are further synthesized into a unified user embedding  $\mathbf{u}_{re}$  for recall by  $\mathbf{u}_{re} = \sum_{i=1}^M \alpha_i \mathbf{v}_i$ . We use a news encoder that is shared with the ranking part to obtain the embedding  $\mathbf{r}^c$  of each candidate news  $D^c$  in the news pool. The final recall relevance score  $\hat{y}_{re}$  between user interest and candidate news is computed by  $\hat{y}_{re} = \mathbf{u}_{re} \cdot \mathbf{r}^c$ .

## 2.3 Model Training

Then we introduce the model training details of *UniRec*. We use a two-stage model training strategy to first learn the ranking part and then learn the recall part. Following prior works (Huang et al., 2013; Wu et al., 2019b,c), we use negative sampling techniques to construct samples for contrastive model learning (Oord et al., 2018). For learning the ranking part, we use clicked news in each impression as positive samples, and we randomly sample  $K$  non-clicked news that are displayed in the same impression as negative samples. The loss function is formulated as follows:

$$\mathcal{L}_{ra} = -\log \left[ \frac{\exp(\hat{y}_{ra}^+)}{\exp(\hat{y}_{ra}^+) + \sum_{i=1}^K \exp(\hat{y}_{ra}^{i-})} \right], \quad (2)$$

where  $\hat{y}_{ra}^+$  and  $\hat{y}_{ra}^{i-}$  denote the predicted click scores of a positive sample and the corresponding  $i$ -th negative sample, respectively. By optimizing this loss function, the parameters of news and user encoders can be tuned. Motivated by (Ying et al., 2018), we fix the news encoder after the ranking model converges. Then, to learn the recall part, we also use clicked news of each user as positive samples, while we randomly select  $T$  non-clicked news from the entire news set as negative samples, which aims to simulate the news recall scenario. The loss function for recall part training is as follows:

$$\mathcal{L}_{re} = -\log \left[ \frac{\exp(\hat{y}_{re}^+)}{\exp(\hat{y}_{re}^+) + \sum_{i=1}^T \exp(\hat{y}_{re}^{i-})} \right], \quad (3)$$

where  $\hat{y}_{re}^+$  and  $\hat{y}_{re}^{i-}$  represent the predicted recall relevance scores of a positive sample and the corresponding  $i$ -th negative sample, respectively.

However, not all basis user embeddings are relevant to the interests of a user. Thus, motivated by Principal Component Analysis (PCA), in the test phase we propose to only use the top  $P$  basis user embeddings with the highest attention weights to compose the user embedding for recall. We denote these basis user embeddings as  $[\mathbf{v}_{t_1}, \mathbf{v}_{t_2}, \dots, \mathbf{v}_{t_P}]$ . We re-normalize their attention weights as follows:

$$\alpha_{t_i} = \frac{\exp(\alpha_{t_i})}{\sum_{j=1}^P \exp(\alpha_{t_j})}. \quad (4)$$

The user embedding  $\mathbf{u}_{re}$  for recall is built by  $\mathbf{u}_{re} = \sum_{i=1}^P \alpha_{t_i} \mathbf{v}_{t_i}$ , which can attend more to the major interests of a user and filter noisy basis user embeddings for better news recall.

## 2.4 Complexity Analysis

We provide some discussions on the computational complexity. In existing news recommendation methods that conduct recall and ranking with separate models, the computational complexity of learning user embeddings for recall and ranking are both  $O(N)$  at least, because they need to encode the entire user behavior sequence. *UniRec* has the same complexity in learning the user embedding for ranking, but the complexity of deriving the user embedding for recall is reduced to  $O(M)$ , where  $M$  is usually much smaller than  $N$ . In addition, the attention network used for synthesizing the user embedding for recall may also be lighter-weight than the user encoder. Thus, the total computational complexity can be effectively reduced.

## 3 Experiments

### 3.1 Dataset and Experimental Settings

We conduct experiments on a large-scale public dataset named MIND (Wu et al., 2020b) for news recommendation. It contains news impression logs of 1 million users on Microsoft News in 6 weeks. The logs in the first five weeks are for training and validation, and the rest logs are for test. The detailed statistics of MIND are shown in Table 1.

# Users	1,000,000	# News	161,013
# Impressions	15,777,377	# Click behaviors	24,155,470
Avg. news title len.	11.52	# Categories	20

Table 1: Statistics of the *MIND* dataset.

In our experiments, following (Wu et al., 2020b) we use news titles to learn news embeddings. The number of basis user embeddings is 20, and they are randomly initialized. The hyperparameter  $P$  that controls the number of basis user embeddings for composing the user embedding for recall in the test phase is 5. The number of negative samples associated with each positive one is 4 and 200 for the ranking and recall tasks, respectively. Adam (Bengio and LeCun, 2015) is used as the optimizer. The batch size is 32. These hyperparameters are selected on the validation set. Following (Wu et al., 2020b), we use AUC, MRR, nDCG@5 and nDCG@10 to evaluate news ranking performance. In addition, we use recall rate of the top 100, 200, 500 and 1000 ranked news to evaluate news recall performance. We repeat every experiment 5 times.

Methods	AUC	MRR	nDCG@5	nDCG@10
EBNR	66.22±0.17	31.97±0.14	34.89±0.17	40.49±0.19
DKN	65.61±0.20	31.58±0.17	34.32±0.19	40.04±0.22
NPA	67.62±0.14	32.69±0.13	35.52±0.15	41.33±0.17
NAML	67.45±0.12	32.48±0.09	35.39±0.10	41.19±0.14
NRMS	68.24±0.09	33.38±0.10	36.34±0.10	42.12±0.13
UniRec	<b>68.41±0.11</b>	<b>33.50±0.10</b>	<b>36.47±0.12</b>	<b>42.26±0.14</b>

Table 2: Ranking performance of different methods.

Methods	R@100	R@200	R@500	R@1000
YoutubeNet	1.395±0.034	2.284±0.039	4.171±0.042	6.867±0.037
Pinnersage	1.431±0.020	2.340±0.018	4.252±0.017	6.927±0.019
Octopus	1.426±0.026	2.392±0.029	4.344±0.031	7.188±0.029
UniRec(all)	1.443±0.023	2.402±0.027	5.022±0.025	8.294±0.026
UniRec(top)	<b>1.516±0.026</b>	<b>2.531±0.024</b>	<b>5.142±0.027</b>	<b>8.485±0.026</b>

Table 3: Recall performance of different methods.

### 3.2 Performance Evaluation

We first compare the ranking performance of *UniRec* with several baseline methods, including: (1) EBNR (Okura et al., 2017), GRU (Cho et al., 2014) network for user interest modeling in news recommendation; (2) DKN (Wang et al., 2018), deep knowledge network for news recommendation; (3) NPA (Wu et al., 2019b), news recommendation with personalized attention; (4) NAML (Wu et al., 2019a), news recommendation with attentive multi-view learning; (5) NRMS (Wu et al., 2019c), news recommendation with multi-head self-attention. The ranking performance of different methods is shown in Table 2. We find that *UniRec* outperforms several compared baseline methods like NAML and NPA. This may be because self-attention has stronger ability in modeling news and user interests. In addition, *UniRec* also slightly outperforms its basic model NRMS. This is because *UniRec* can capture the orders of words and behaviors via position embedding.

In the news recall task, we compare *UniRec* with top basis user embeddings (denoted as *UniRec(top)*) with the following baseline methods: (1) YoutubeNet (Covington et al., 2016), using the average of clicked news embeddings for recall; (2) Pinnersage (Pal et al., 2020), an item recall method based on hierarchical clustering; (3) Octopus (Liu et al., 2020), learning elastic number of user embeddings for item recall; (4) UniRec(all), a variant of *UniRec* that uses all basis user embeddings to compose the user embedding for recall. We

show the recall performance of different methods in Table 3. We find YoutubeNet is less performant than other recall methods. This may be because different user behaviors may have different importance in user interest modeling and simply average their embeddings may be suboptimal. In addition, both *UniRec(top)* and *UniRec(all)* outperform other baseline methods. This is because our approach can exploit the user interest information inferred from the ranking module to enhance news recall. In addition, our approach is a unified model for both recall and ranking, which has better efficiency in online systems than other methods. Besides, *UniRec(top)* outperforms its variant *UniRec(all)*. It may be because selecting the basis user embeddings with top attention weights can learn accurate user interest embeddings by attending to major user interests and filtering noisy ones. The above results validate the effectiveness of our method in both news ranking and recall.

### 3.3 Case Study

We verify the effectiveness of *UniRec* in news recall via several case studies. Fig. 3 shows the clicked news of a random user and several top news recalled by *UniRec*. From the user’s clicked news, we can infer that this user may be interested in finance, sports and TV shows. We find the recall result of *UniRec* covers user interest categories of clicked news, but also keeps some diversity with them. It shows that *UniRec* can generate accurate and diverse personalized news recall results.



	Category	Title
Clicked News	Finance	Chipotle customers say the chain is charging them hundreds of dollars in fake orders
	Sports	Every touchdown from every game in week 9
	TV	fresh off the boat canceled after six seasons
UniRec Recall	Sports	The Patriots opened with a grinding 16-play drive in which nearly everything went right
	Finance	Dean foods files for bankruptcy
	TV	Viral Wheel of Fortune Contestant and His Wife Clarify Hilarious 'Loveless Marriage' Intro
	TV	8 of the best and 8 of the worst tv shows that got canceled this year, so far
	Sports	Browns, Steelers brawl at end of cleveland's 21-7 win

Figure 3: The news clicked by a randomly sampled user and the top news recalled by *UniRec*.

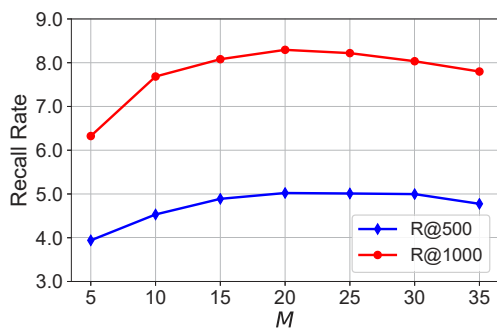


Figure 4: Influence of the basis user embedding number.

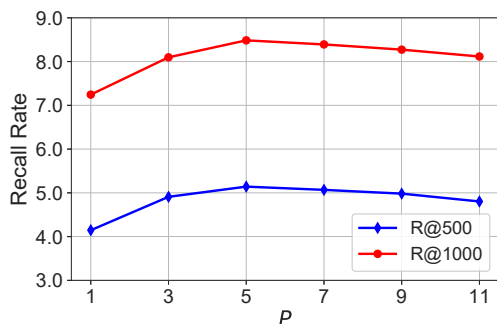


Figure 5: Influence of the hyperparameter  $P$ .

### 3.4 Hyperparameter Analysis

Finally, we study the influence of two important hyperparameters in our *UniRec* method, including the total number  $M$  of basis user embeddings and the number  $P$  of basis user embeddings for composing the user embeddings for recall. We first set  $P = M$  and tune the value of  $M$ . The recall performance is shown in Fig. 4. We find the performance is suboptimal when  $M$  is too small, which may be due to the diverse user interests cannot be covered by a few basis user embeddings. However, the performance also descends when  $M$  is large.

This may be because it is difficult to accurately select informative basis user embeddings for user interest modeling. In addition, the computation and memory costs also increase. Thus, we set  $M$  to a medium value (i.e., 20) that yields the best performance. We then tune the value of  $P$  under  $M = 20$ . The results are shown in Fig. 5. We find the performance is suboptimal when  $P$  is very small. This is intuitive because the user interests cannot be fully covered. However, the performance also declines when  $P$  is relatively large. This may be because basis user embeddings with relatively low attention weights are redundant or even noisy for user interest modeling. Thus, we choose to use 5 basis user embeddings to compose the user embedding for recall.

## 4 Conclusion

In this paper, we present a unified approach for recall and ranking in news recommendation. In our method, we first infer a user embedding for ranking from historical news click behaviors via a user encoder model. Then we derive a user embedding for recall from the obtained user embedding for ranking by regarding it as attention query to select a set of basis user embeddings that encode different general user interests. Extensive experiments on a benchmark dataset validate the effectiveness of our approach in both news ranking and recall.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant numbers U1936208, U1936216, and 61862002, and the research initiation project of Zhejiang Lab (No. 2020LC0PI01).

## References

- Yoshua Bengio and Yann LeCun. 2015. Adam: A method for stochastic optimization. In *ICLR*.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734.
- Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *RecSys.*, pages 191–198. ACM.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.
- Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW*, pages 278–288.
- Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph enhanced representation learning for news recommendation. In *WWW*, pages 2863–2869.
- Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, pages 2333–2338. ACM.
- Wang-Cheng Kang and Julian McAuley. 2019. Candidate generation with binary codes for large-scale top-n recommendation. In *CIKM*, pages 1523–1532.
- Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. 2018. News recommender systems—survey and roads ahead. *Information Processing & Management*, 54(6):1203–1227.
- Zheng Liu, Jianxun Lian, Junhan Yang, Defu Lian, and Xing Xie. 2020. Octopus: Comprehensive and elastic user representation for the generation of recommendation candidates. In *SIGIR*, pages 289–298.
- Zheng Liu, Yu Xing, Fangzhao Wu, Mingxiao An, and Xing Xie. 2019. Hi-fi ark: Deep user representation via high-fidelity archive network. In *IJCAI*, pages 3059–3065.
- Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *TPAMI*, 42(4):824–836.
- Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *KDD*, pages 1933–1942. ACM.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Aditya Pal, Chantat Eksombatchai, Yitong Zhou, Bo Zhao, Charles Rosenberg, and Jure Leskovec. 2020. Pinnersage: Multi-modal user embedding framework for recommendations at pinterest. In *KDD*, pages 2311–2320.
- Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021a. Personalized news recommendation with knowledge-aware interactive matching. In *SIGIR*, pages 61–70.
- Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021b. Pp-rec: News recommendation with personalized user interest and time-aware news popularity. In *ACL*, pages 5457–5467.
- Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2021c. Uni-fedrec: A unified privacy-preserving news recommendation framework for model training and online serving. In *EMNLP Findings*, pages 1438–1448.
- Tao Qi, Fangzhao Wu, Chuhan Wu, Peiru Yang, Yang Yu, Xing Xie, and Yongfeng Huang. 2021d. Hierec: Hierarchical user interest modeling for personalized news recommendation. In *ACL*, pages 5446–5456.
- Qiaoyu Tan, Ninghao Liu, Xing Zhao, Hongxia Yang, Jingren Zhou, and Xia Hu. 2020. Learning to hash with graph neural networks for recommender systems. In *WWW*, pages 1988–1998.
- Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *ACL*, pages 836–845.
- Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. Dkn: Deep knowledge-aware network for news recommendation. In *WWW*, pages 1835–1844.
- Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019a. Neural news recommendation with attentive multi-view learning. In *IJCAI*, pages 3863–3869.
- 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.
- Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019c. Neural news recommendation with multi-head self-attention. In *EMNLP*, pages 6390–6395.
- Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2021a. Personalized news recommendation: A survey. *arXiv preprint arXiv:2106.08934*.

- Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2021b. User-as-graph: User modeling with heterogeneous graph pooling for news recommendation. In *IJCAI*, pages 1624–1630.
- Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020a. User modeling with click preference and reading satisfaction for news recommendation. In *IJCAI*, pages 3023–3029.
- Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021c. Feedrec: News feed recommendation with various user feedbacks. *arXiv preprint arXiv:2102.04903*.
- Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020b. Mind: A large-scale dataset for news recommendation. In *ACL*, pages 3597–3606.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL-HLT*, pages 1480–1489.
- Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*, pages 974–983.