RMBERT: News Recommendation via Recurrent Reasoning Memory Network over BERT

Qinglin Jia, Jingjie Li*, Qi Zhang, Xiuqiang He, Jieming Zhu
*Corresponding author
Noah's Ark Lab, Huawei, China
{jiaqinglin2,lijingjie1,zhangqi193,hexiuqiang1,jamie.zhu}@huawei.com

ABSTRACT

Personalized news recommendation aims to alleviate information overload and help users find news of their interests. Accurately matching candidate news and users' interests is the key to news recommendation. Most existing methods separately encode each user and news into vectors by news contents and then match the two vectors. However, a user's interest may differ in each news or each topic of one news. It's necessary to dynamically learn user and news vector and model their interaction. In this work, we present Recurrent **R**easoning **M**emory Network over **BERT** (RMBERT) for news recommendation. Compared with other methods, our approach can leverage the ability of content modeling from BERT. Moreover, the recurrent reasoning memory network which performs a series of attention based reasoning steps can dynamically learn user and news vector and model their interaction in each step. As a result, our approach can better model user's interests. We conduct extensive experiments on a real-world news recommendation dataset and the results show that our approach significantly outperforms existing state-of-the-art methods.

CCS CONCEPTS

• **Information systems** → **Personalization**; **Recommender systems**; • **Computing methodologies** → **Knowledge representation and reasoning**.

KEYWORDS

News Recommendation; Deep Neural Network; Attention Model

ACM Reference Format:
Qinglin Jia, Jingjie Li*, Qi Zhang, Xiuqiang He, Jieming Zhu. 2021. RMBERT: News Recommendation via Recurrent Reasoning Memory Network over BERT. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463234

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463234

1 INTRODUCTION

Due to the rapid development of the Internet, people's reading habits have gradually shifted to online news platforms [8]. Compared with the traditional media such as news paper and TV, online news platforms such as MSN News[1] and Google News[2] can aggregate news from various sources and distribute news articles to specific users. Personalized recommender systems plays a pivotal role in helping users quickly discover news of their interests.

Generally, news recommendation systems need to model news contents precisely. This is because news articles are highly time-sensitive and out of date news are quickly substituted by newer ones, that makes the ID-based features less effective [13]. Traditional news recommendation methods rely on manual feature engineering to build news and user representation, such as Wide &Deep [2], DeepFM [6], DCN [14], etc. Recently, some methods explore to use neural networks to learn news and user representation in an end-to-end manner and optimize these representations during the training process. DKN [13], NPA [15] and LSTUR [1] proposed to use CNN to learn news representation and then match the candidate news and the user's clicked news. DKN incorporated knowledge graph representation into news recommendation. NPA enhanced news and user representation by introducing the embeddings of user's ID for attending to important words and news. LSTUR combined long- and short-term user representation to model user more precisely. NRMS [16] introduced multi-head self-attention to learn news representation. FIM [12] extracted multilevel representations for each news via stacked dilated convolutions.

Despite the remarkable gains delivered by them, we would argue that those methods are still insufficient n capturing the rich semantics in textual contents. Most existing methods take CNNs or GRU as news encoder, and are typically trained with in-domain news texts. This is insufficient to obtain a comprehensive news representation, and will further hurt the recommendation performance. Inspired by the recent milestone work on pretrained language model BERT[3], we adopt BERT to encode the news texts. BERT employs deep transformers to enhance language understanding from large-scale unlabeled corpora. The pretrained model BERT can produce more expressive news embeddings, We independently encode each news to allow to precompute news embeddings offline and greatly reducing the computational load during online inference.

Moreover, there are two common observations in the news recommendation scenario. First, different users may be interested in different parts of the candidate news. Second, user's interests are usually diverse, and the different news in the user history represent the different interests of this user. Previous work [9, 13, 16] using

[1]https://www.bing.com/news
[2]https://news.google.com/

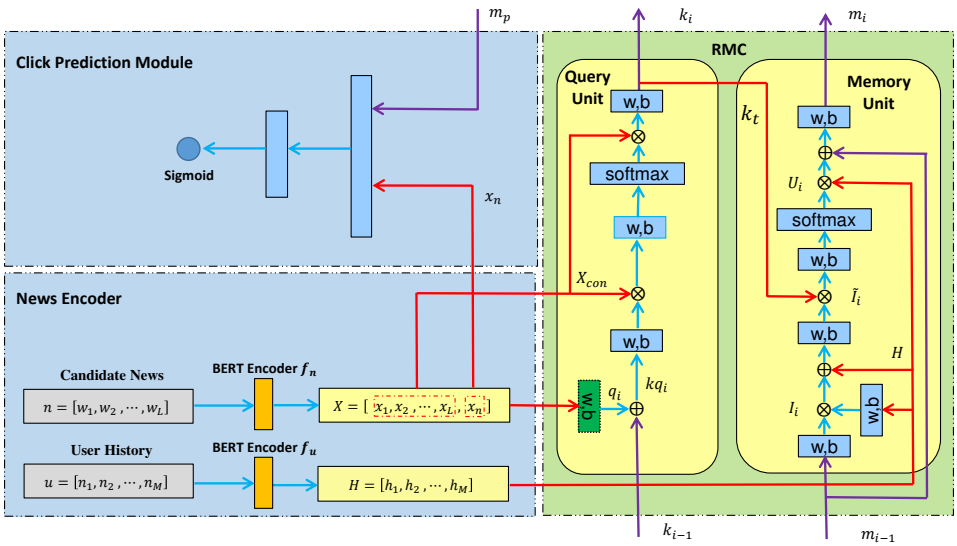


Figure 1: Architecture of RMBERT model. The recurrent reasoning memory network contains p RMCs, and they work in tandem. All RMCs share the same parameters.

one fixed embedding vector to represent user and news may limit the expressiveness of the model. As a solution, we need to dynamically determine the user vector with respect to the candidate news. The memory network can capture higher order complex relations between two objects and has made great strides in many research areas such as question answering [4, 7], sentiment analysis [11], machine comprehension [10], etc. Motivated by its success, we design a recurrent structure for the iterative reasoning processes that several reasoning memory cells (RMC) are cascaded to perform structured reasoning operations step by step to determine the matching score.

We term the resulting model as Recurrent Reasoning Memory Network over BERT, RMBERT for brevity. The architecture of our approach is shown in Figure 1. The news encoder module digests the candidate news and some user's clicked news by processing the title of each news and generating the news embeddings. After that, the recurrent reasoning memory network contains some RMCs, which takes news embeddings as input and work in tandem to perform inference procedures step by step. In each step, RMC performs reasoning operations by a query unit and a memory unit. The query unit uses attention mechanism to select some aspects of the candidate news embedding that the user may be interested in to update the query state. The memory unit dynamically retrieves the user's interests from the user clicked news embeddings according to the query state. After several reasoning steps, RMBERT can capture the user's preference for the candidate news from multiple aspects. At last, we recommend news according to the last memory state which contains the interaction information between user and candidate news.

The main contributions of this work are summarized as follows.

(1) We independently encode the candidate news and the user clicked news using BERT. Our approach can leverage the expressiveness of BERT while simultaneously gaining the ability to precompute news embeddings offline.

(2) We propose a recurrent structure to perform attention based reasoning operations step by step. In each step, the RMC can infer the matching score between some parts of candidate news and user's interests. RMC can dynamically determine news and user vector in each step and explore their matching in many aspects.

(3) Extensive experiments on a real-world dataset collected from MSN news validate that our approach achieves better performance than the existing state-of-the-art methods.

2 PROPOSED METHOD

2.1 News Encoder

We use n to denote the candidate news and $u = [n_1, n_2, \cdots, n_M]$ to denote the user history, where M is the user history length. f_n encodes n into a bag of fixed-size news word embeddings X while f_u encodes each news n_i in u into another bag X_i, f_n and f_u are pretrained BERT model and share the same parameters. First, we tokenize news title of n into its BERT-based WordPiece tokens $w_1, w_2, \cdots, w_L$, where L is the news title sequence length. We follow the approach by BERT: prepend the [CLS] token to the beginning of the input tokens. The output embedding corresponding to [CLS] token is used as the aggregate sequence representation that carries global information. Then, this sequence of input tokens is passed into BERT, which computes a contextualized representation of each token.

We use the final output of [CLS] token as the candidate news embedding x_n. The output of word token list is the contextual word embeddings of the candidate news $X_{con} = [x_1, x_2, \cdots, x_L]$. In order to reserve the original word-level information of the candidate news, we retain the contextual word embeddings of the candidate news X_{con} and input it into query unit. Similarly, for each news in user history, we first prepend BERT's start token [CLS], then pass the sequence of input tokens into BERT. We use h_i to denote the news embedding of news n_i in u. The user's clicked news embedding set H can be defined as $H = [h_1, h_2, \cdots, h_M]$. The candidate news embedding x_n , the contextual word embeddings of candidate news X_{con} and the user's clicked news embedding set H are fed into the

reasoning memory network to estimate the matching score of n to u.

2.2 Reasoning Memory Cell

Previous work learns fixed candidate news vector. This will constrain the expressiveness of model. In our work, we designed a recurrent structure contains p RMCs. In each RMC, The query unit learns different candidate news vector to represent different aspects of the candidate news and then the memory unit matches user's interests. These cells are stringed together to perform a series of matching procedure, and share the same parameters. Our model can explore the matching signal in many aspects. The i-th cell receives the previous cell's dual hidden states, the candidate news word embeddings X and the user's clicked news embedding set H as input and update the dual hidden states k_i and m_i. At the beginning, k_0 is initialized by x_n and m_0 is randomly initialized.

2.2.1 The Query Unit. The query unit is designed to determine the key information that the memory unit should consider at each step i. The query unit utilizes the previous query state k_{i-1} and the candidate news word embeddings X to update the query state k_i which contains the key information of the candidate news .

To focus on different aspects of the candidate news at different iteration steps, we utilize different W_q^i and b_q^i to linearly transform the candidate news embeddings into a position-aware vector q_i. Then, we combine q_i with the previous query state k_{i-1} and via a linear transformation to generate the vector kq_i. So cells can decide the aspects that should focus on.

$$q_i = W_p^i \cdot x_n + b_p^i \tag{1}$$

$$kq_i = W_k \cdot [q_i, k_{i-1}] + b_k \tag{2}$$

where $W_p^i \in \mathbb{R}^{d\times d}$, $b_p^i \in \mathbb{R}^d$, $W_k \in \mathbb{R}^{(d+d_q)\times d}$, $b_k \in \mathbb{R}^d$ are projection parameters, i is the iteration step, d_q is the dimension of the query unit , d is the word embedding dimension.

During the reasoning process, we still need to restrict the space of valid reasoning operations to prevent the query unit from diverging. We apply an attention network to sure that the query unit still focuses on the news words. Finally, we sum the contextual word embeddings according to the attention weight.

$$k_i = W_q \cdot \sum_{j=1}^{L} softmax(W_\alpha \cdot (kq_i \odot X_{con}) + b_\alpha) \cdot X_{con} + b_q \tag{3}$$

where $W_\alpha \in \mathbb{R}^{d\times 1}$, $b_\alpha \in \mathbb{R}^1$, $W_q \in \mathbb{R}^{d\times d_q}$, $b_q \in \mathbb{R}^{d_q}$ are projection parameters, $\odot$ is the element-wise product and k_i is the current query state. Compared with traditional attention networks, we abandon the constraint of $\sum_{j=1}^{L} \alpha_i = 1$ to preserve the intensity of word informativeness.

2.2.2 The Memory Unit. The memory unit receives the previous memory state m_{i-1}, the current query state k_i and the user's clicked news embedding set H as input and perform a reasoning operation to determine the user's interests in the candidate news.

To determine the user's representation from the user's clicked news, we perform a two-stage process over the user's clicked news. First, we apply a linearly transformation to m_{i-1}. For the user's clicked news embeddings, we adopt the same processing. We can get the interaction I_i between them by computing the element-wise product. Because m_{i-1} contains the prior extracted matching signal between the user's clicked news and the candidate news. The memory unit can inherit this relevance step by step when performing reasoning step. Then, we combine the vector I_i and H through a linear transformation to consider new information from H.

$$I_i = [W_r \cdot m_{i-1} + b_r] \odot [W_H \cdot H + b_H] \tag{4}$$

$$\widetilde{I}_i = W_I [I_i, H] + b_I \tag{5}$$

where $W_r \in \mathbb{R}^{d_m\times d}$, $b_r \in \mathbb{R}^d$, $W_H \in \mathbb{R}^{d\times d}$, $b_H \in \mathbb{R}^d$, $W_I \in \mathbb{R}^{2d\times d}$ $b_I \in \mathbb{R}^d$ are projection parameters and d_m is the dimension of memory unit. $\widetilde{I}$ is the user representation in current step.

Due to the diversity of users' interests, different news represents different user's interests. We compute the similarly between the query state k_i and the user's representation $\widetilde{I}_i$ by an attention network. The user's interests in the candidate news U_i is the summation of the representations of the user's clicked news H weighted by their attention weight.

$$U_i = \sum_{j=1}^{M} softmax(W_\beta \cdot \left(\widetilde{I}_i \odot k_i\right) + b_\beta) \cdot H \tag{6}$$

where $W_\beta \in \mathbb{R}^{d\times 1}$, $b_\beta \in \mathbb{R}^1$ are parameters of attention network.

The memory state holds the user's preference for the candidate news obtained from the interaction between the candidate news and the user's clicked news. To update the memory state, we combine the previous memory state m_{i-1} with the current user's interests information U_i and via a linear transformation.

$$m_i = W_m [m_{i-1}, U_i] + b_m \tag{7}$$

where $W_m \in \mathbb{R}^{(d_m+d)\times d_m}$, $b_m \in \mathbb{R}^{d_m}$ are projection parameters.

After p reasoning steps, the recurrent reasoning memory network has obtained the relation information between the user history and the candidate news.

2.3 Click Prediction Module

We predict the probability of a user clicking the candidate news based on the last memory state m_p. We combine the candidate news embedding x_n with m_p and pass them through two hidden layers. The probability of the user clicking the candidate news is predicted by the following equations:

$$\begin{aligned} o &= ReLU\left(W_1 \cdot \left[x_n, m_p\right] + b_1\right) \\ \hat{y} &= Sigmoid\left(W_2 \cdot o + b_2\right) \end{aligned} \tag{8}$$

where W_1, W_2, b_1 and b_2 are parameters of dense layers. RMBERT is trained with BCEWithLogits loss.

3 EXPERIMENT

3.1 Datasets and Experiment Settings

3.1.1 Datasets. We evaluate our approach using a publicly accessible dataset MIND [17], which was collected from the user behavior logs of Microsoft News. There are two versions of the dataset named MIND-large and MIND-small. The MIND-large contains 3 million users and the MIND-small randomly sample 50,000 users. An impression log records the news that are displayed to a user when he visits the news platform at a specific time and the history click

Table 1: Performance comparison of different models.

	MIND-small			MIND-large		
Methods	AUC	MRR	nDCG@5	AUC	MRR	nDCG@5
DeepFM	0.5752	0.2391	0.2489	0.6152	0.2759	0.2934
DCN	0.5801	0.2393	0.2519	0.6124	0.2679	0.2857
DKN	0.6290	0.2837	0.3099	0.6715	0.3206	0.3531
NRMS	0.6563	0.3096	0.3413	0.6785	0.3297	0.3632
LSTUR	0.6587	0.3078	0.3395	0.6801	0.3290	0.3629
NPA	0.6465	0.3001	0.3314	0.6752	0.3261	0.3581
FIM	0.6531	0.3123	0.3400	0.6852	0.3344	0.3704
RMBERT	**0.6746**	**0.3249**	**0.3614**	**0.6956**	**0.3410**	**0.3775**

behavior on these displayed news. We use the logs in the first 6 days for training and the last day for testing. We further randomly sampled 10% of training samples for validation.

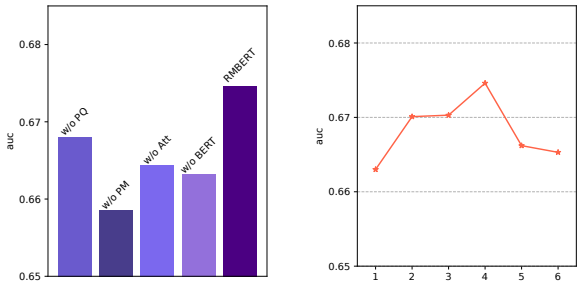
3.1.2 Experiment Setup. We compare our approach against several competitive baselines including DCN [14], DeepFM [6], DKN [13], NRMS [16], LSTUR [1], NPA [15], FIM [12]. We use the handcrafted features as the complement of ID features for DeepFM and DCN. The handcrafted features for news texts are TF-IDF features which are extracted from news title. The performance is judged by the AUC, MRR and nDCG@5 scores over all impressions. Most of the baseline models are implemented in TensorFlow, based on the public code of Recommender [5], following their parameter settings. We tune other hyperparameters based on validation set. We initialize the BERT based news eocoder using Google's official pretrained model. The length of recurrent reasoning memory network is 4. We fine-tune RMBERT using Adam with learning rate $2 \times e^{-5}$and batch size is set to 50.

3.2 Results and Discussion

The overall performance of all comparing methods are summarized in Table 1. The best and second results are highlighted in boldface and underlined respectively.

We have several observations. First, the performance of *DeepFM* and *DCN* is consistently lower than other models. This is because these handcrafted features are fixed and can not be optimized during the training process. Second, *NRMS* performs better than *DKN* and *NPA*. This is probably because *self-attention* can capture long-range contexts of words, while CNN can only model local contexts. Third, these models *NRMS, LSTUR, NPA, FIM* which use attention mechanism to learn news and user representation in an end-to-end manner perform better than other baseline models. This may be explained by the fact that attention mechanism can distinguish the informative words in news titles and it can select some news which can reflect the preference of users.

In all, our approach can consistently outperform all compared baseline models in terms of all metrics. We attribute the superiority of RMBERT to its three properties: (1) RMBERT uses BERT to capture long-range contexts of words. Moreover, the pretrained model can leverage the out-domain unlabeled large-scale corpora, which can generate better news embeddings. (2) RMBERT uses word-level attention in query unit and news-level attention in memory unit to dynamically determine news and user vectors in each step. (3) The recurrent structure can iteratively select information from the candidate news and match the user's clicked news, which can enhance model capability.



(a) Variant models (b) The number of RMCs

Figure 2: The effectiveness of different modules on MIND-small dataset.

We conduct abalation studies to explore the effectiveness of different parts in our approach. We design 4 variants of RMBERT to understand the contribution of RMC and BERT encoder: (1) ignore previous query state in query unit (*w/o PQ*) (2) ignore previous memory state in memory unit (*w/o PM*) (3) replace attention network with average pooling (*w/o Att*) (4) replace BERT with Bi-LSTM (*w/o BERT*). Figure 2 (a) presents experimental results with above variants. RMBERT achieves the best performance. The results prove that RMBERT can benefit a lot from pretrained model BERT. The performance of *w/o Att* indicates that highlighting the important words and news using the attention network can help model understand user's interests. The query unit can only learn a few aspects from the candidate news at a time and it need to understand the whole candidate news step by step. As to *w/o PM*, this variant shows the worst results among all variants. This is because the memory unit can only superficially comprehend the relatedness between two objects at one step. This demonstrates the effectiveness of our recurrent structure which can decompose the complex inference procedure into a series of simple operations.

We further investigate how RMBERT performs with different number of RMC ranging from 1 to 6. Figure 2 (b) shows the auc score with different number of RMC. We can observe that our model achieves the best performance in terms of AUC, when the number of RMC is 4. The AUC score generally increases as the number of RMC gets larger, since more RMCs can be able to perform more reasoning steps and enhance model capability. However, the trend changes when $p = 5$ due to probable overfitting. More RMCs will increase the complexity of our model, and consume more training time. It's important to select a suitable p for balancing the performance and the the complexity of model.

4 CONCLUSION

In this paper, we propose a novel model for news recommendation that employs the recurrent reasoning memory network over BERT. By independently encoding candidate news and user's clicked news with BERT, RMBERT can leverage the expressiveness of BERT. In addition, our approach can dynamically capture user's preference to the candidate news by performing a series of reasoning operations. As a result, the news recommendation lists can be extended reasonably. The experiments on the real-world dataset collected from MSN news demonstrate the effectiveness of our model.

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 *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 336–345.

[2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] Jinlan Fu, Yi Li, Qi Zhang, Qinzhuo Wu, Renfeng Ma, Xuanjing Huang, and Yu-Gang Jiang. 2020. Recurrent Memory Reasoning Network for Expert Finding in Community Question Answering. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 187–195.

[5] Scott Graham, Jun-Ki Min, and Tao Wu. 2019. Microsoft recommenders: tools to accelerate developing recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 542–543.

[6] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[7] Drew A Hudson and Christopher D Manning. 2018. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067* (2018).

[8] Talia Lavie, Michal Sela, Ilit Oppenheim, Ohad Inbar, and Joachim Meyer. 2010. User attitudes towards news content personalization. *International journal of human-computer studies* 68, 8 (2010), 483–495.

[9] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1933–1942.

[10] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1047–1055.

[11] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900* (2016).

[12] Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained Interest Matching for Neural News Recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 836–845.

[13] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*. 1835–1844.

[14] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.

[15] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2576–2584.

[16] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with multi-head self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 6390–6395.

[17] 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3597–3606.