Combining Explicit Entity Graph with Implicit Text Information for News Recommendation

Xuanyu Zhang
Du Xiaoman Financial
Beijing, China
zhangxuanyu@duxiaoman.com

Qing Yang
Du Xiaoman Financial
Beijing, China
yangqing@duxiaoman.com

Dongliang Xu
Du Xiaoman Financial
Beijing, China
xudongliang@duxiaoman.com

ABSTRACT

News recommendation is very crucial for online news services to improve user experience and alleviate information overload. Precisely learning representations of news and users is the core problem in news recommendation. Existing models usually focus on implicit text information to learn corresponding representations, which may be insufficient for modeling user interests. Even if entity information is considered from external knowledge, it may still not be used explicitly and effectively for user modeling. In this paper, we propose a novel news recommendation approach, which combine explicit entity graph with implicit text information. The entity graph consists of two types of nodes and three kinds of edges, which represent chronological order, related and affiliation relationship. Then graph neural network is utilized for reasoning on these nodes. Extensive experiments on a real-world dataset, Microsoft News Dataset (MIND), validate the effectiveness of our proposed approach.

CCS CONCEPTS

• Information systems → Information systems applications; **• Human-centered computing → Collaborative and social computing**; **• Social and professional topics → User characteristics**; **• Applied computing → Sociology**.

KEYWORDS

News Recommendation, Graph Neural Networks, Entity Graph

ACM Reference Format:
Xuanyu Zhang, Qing Yang, and Dongliang Xu. 2021. Combining Explicit Entity Graph with Implicit Text Information for News Recommendation. In *Companion Proceedings of the Web Conference 2021 (WWW '21 Companion), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3442442.3452329

1 INTRODUCTION

Recently, online news platforms, such as Google News [1], Yahoo News [2] and MSN News[3], have gradually attracted people's attention. Because a large volume of news are generated from the worldwide everyday, personalized news recommendation system plays an increasingly important role. It can help users to find their interested news from an overwhelming number of news. Therefore, news recommendation has become a crucial technology in academic and industry to alleviate information overload and improve users' reading experiences.

There are many deep learning based methods [9, 11, 15, 16, 18] proposed for personalized news recommendations. In these models, how to learn news and user representation is a worthwhile problem. Existing methods usually focus on implicit text information to obtain corresponding representations. For instance, Okura et al. [11] proposed to learn news representations from the body of news articles via auto-encoders, and then learn user representations from news representations by applying GRU to their browsed news. Wu et al. [16] proposed to learn unified news representations from titles, bodies and topic categories via an attentive multi-view learning model, and then learn user representations by using attention mechanism to select informative news. Even if Wang et al. [15] and Liu et al. [9] proposed to learn knowledge-aware news representation from news articles and external knowledge, the interaction between different news may still be insufficient for learning user representations. Because the entity information is summarized implicitly in the dense news vectors, which can not be used explicitly across news for modeling user interests. Besides, entity embedding extracted from knowledge graph will take up extra space in real scenarios.

Thus, in this paper, we propose to use graph neural networks (GNN) to explicitly model user interests among different entities and different news. And we combine the explicit entity graph with implicit text information to further enhance user representation. There are two types of nodes in our proposed entity graph, i.e., news node and entity node. The representation of news node comes from news extractor in our model. And the representation of entity node is from corresponding word embeddings. The edges between different news nodes and the edges between news and entity nodes in the same news are unidirectional, which represent chronological order and affiliation relationship, respectively. The edges between entity nodes are bidirectional, which connects the same entity in different news. Besides, mentioned by ResNet [3], final news representation is composed of original implicit news representation and reasoned news representation in explicit entity graph. By integrating traditional text information and entity graph mentioned above, user interests can be captured more precisely and completely. We conduct extensive experiments on a real-world dataset, Microsoft News Dataset (MIND). The results show our approach can effectively improve the performance of news recommendation, achieving state-of-the-art results.

[1]https://news.google.com/
[2]https://news.yahoo.com/
[3]https://www.msn.com/en-us/news

WWW '21 Companion, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442442.3452329

2 RELATED WORK

Personalized news recommendation can effectively improve user reading experience. Traditional news recommendation methods [2, 10, 13] mainly utilize manual feature engineering to build news and user representations. For instance, Bayesian model is used to generate categories and interest features for news representation [10]. Explicit Localized Semantic Analysis (ELSA) is proposed to extract topic and location features from Wikipedia pages for location-based news recommendation [13]. And deep fusion model (DMF) is proposed to fuse various handcrafted features for news representation [8]. However, these methods usually need expertise in specific domain for feature engineering.

Recently, deep learning based methods have also gained a lot of attention and achieved better performance. For example, Okura et al. [11] proposed to learn news representations from the body of news articles via auto-encoders, and then learn users representations from news representations by applying GRU to their browsed news. Wu et al. [18] proposed to use multi-head attention to learn news representations from news title by modeling the interactions between words and learn user representations by capturing the relatedness between the browsed news. An et al. [1] proposed to learn both long- and short-term user representations via GRU network. However, these methods only exploit implicit text information to learn corresponding representations, which may be insufficient for modeling news and users. Different from these methods, our approach focuses on explicit entities in user modeling to learn different relationships among news and entities. It integrates explicit entity graph with implicit text information.

Meanwhile, with the development of deep learning, graph neural networks (GNN) are widely used in natural language processing research, including text classification, machine translation and question answering [21]. In news recommendation, graph neural networks are also adopted. For example, Hu et al. [4] proposed to construct a heterogeneous user-news-topic graph with long-term and short-term user interest to explicitly model the interactions among users, news and topics with complete historic user clicks. And Hu et al. [5] proposed to encode high-order relationships into user and news representations by information propagation along the graph. However, users are viewed as nodes in these models, which makes the calculation of graph neural network more complicated. And the initial representation of users cannot obtained from graph. Different from these GNN methods, in this work, we only rely on news contents in user modeling. Explicit entities and implicit text information are all extracted from news articles. User nodes are not involved in the graph. The representations of users are obtained by aggregating the original news representations and reasoned news nodes. Experiments on a real-world dataset, MIND, validate our approach can learn better news and user representations, and achieve better performance on news recommendation than existing methods.

3 PROBLEM DEFINITION

The news recommendation problem in our paper can be illustrated as follows. Given a candidate news d_c and a user u with Q clicked news d_h, the model aims to predict whether the user u will click the candidate news. Then recommendations are given based on the ranking of news-user pair scores. Both d_c and d_h can be unified as news d, which contains lots of attributes, including titles, abstract, body and entities. The title with M tokens is $[w_1^t, w_2^t, ..., w_M^t]$. The abstract with N tokens is $[w_1^a, w_2^a, ..., w_N^a]$. The body with O tokens is $[w_1^b, w_2^b, ..., w_O^b]$. And the news with L entities is $[w_1^e, w_2^e, ..., w_L^e]$.

4 APPROACH

In this section, we will present our approach from three stages. First, we calculate implicit text representation based on different attributes of news. Then, we use GNN to reason on explicit entity graph and implicit text information. Finally, we predict the click probability according to news and user representations, which integrates the results of the first two stages. The architecture of our model is shown in Fig. 1.

4.1 Implicit Text Information

Implicit semantic features of news can be extracted directly from the text in title, abstract and body. We use three layers to obtain the representation for each component of news. The first layer is word embedding layer, which can convert each word to corresponding distributed representation. Then in the second layer, we use convolutional neural networks (CNNs) to extract implicit text features further.After these two layers, the sequence of title, abstract and body can be represented as $[\mathbf{w}_1^t, \mathbf{w}_2^t, ..., \mathbf{w}_M^t]$, $[\mathbf{w}_1^a, \mathbf{w}_2^a, ..., \mathbf{w}_N^a]$ and $[\mathbf{w}_1^b, \mathbf{w}_2^b, ..., \mathbf{w}_O^b]$, respectively. Following [16], we use attention mechanism for aggregation in the third layer. The weights of each word in title, abstract and body can be formulated as:

$$\begin{aligned} c_i^t &= \mathbf{q}_t^T \tanh(\mathbf{V}_t \times \mathbf{w}_i^t + \mathbf{v}_t), \\ \alpha_i^t &= \frac{e^{c_i^t}}{\sum_{j=1}^{M} e^{c_j^t}} \end{aligned} \quad (1)$$

$$\begin{aligned} c_i^a &= \mathbf{q}_a^T \tanh(\mathbf{V}_a \times \mathbf{w}_i^a + \mathbf{v}_a), \\ \alpha_i^a &= \frac{e^{c_i^a}}{\sum_{j=1}^{N} e^{c_j^a}} \end{aligned} \quad (2)$$

$$\begin{aligned} c_i^b &= \mathbf{q}_b^T \tanh(\mathbf{V}_b \times \mathbf{w}_i^b + \mathbf{v}_b), \\ \alpha_i^b &= \frac{e^{c_i^b}}{\sum_{j=1}^{O} e^{c_j^b}} \end{aligned} \quad (3)$$

where $\mathbf{V}_t$, $\mathbf{V}_a$, $\mathbf{V}_b$, $\mathbf{v}_t$, $\mathbf{v}_a$ and $\mathbf{v}_b$ are the projection parameters, $\mathbf{q}_t$, $\mathbf{q}_a$ and $\mathbf{q}_b$ denote the attention query vector to be learned [17]. Then we can obtain the dense representation of title, abstract and body ($\mathbf{d}_t$,$\mathbf{d}_a$ and $\mathbf{d}_b$) by aggregating the sequence of word embedding:

$$\begin{aligned} \mathbf{d}_t &= \sum_{i=1}^{M} \alpha_i^t \mathbf{w}_i^t, \\ \mathbf{d}_a &= \sum_{i=1}^{N} \alpha_i^a \mathbf{w}_i^a, \\ \mathbf{d}_b &= \sum_{i=1}^{O} \alpha_i^b \mathbf{w}_i^b \end{aligned} \quad (4)$$

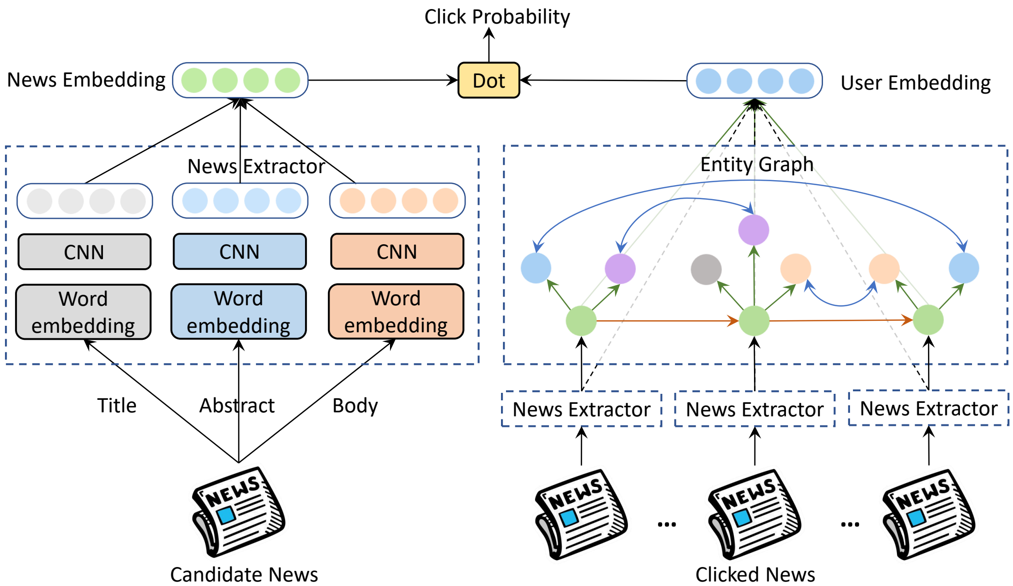

Figure 1: The framework of our model. In the entity graph, green nodes denote news, which are learned from news extractor. Nodes in other colors represent entities. (Best see in colors)

Similarly, we can obtain the news representation by aggregating the representation of abstract, title and body, i.e., $\mathbf{d} = \sum \alpha_i \mathbf{d}_i$, where $i \in \{t, a, b\}$, α_i is the attention weights for each component. The implicit representation of news will be used in entity graph for user representations.

4.2 Explicit Entity Graph

This part aims to reason over explicit entity and implicit text by graph neural networks. First, the entity sequence in news is converted to corresponding vectors $[\mathbf{w}_1^e, \mathbf{w}_2^e, ..., \mathbf{w}_L^e]$ according to the same word embedding used in implicit text information. The entities are provided by the MIND dataset, which are extracted and linked to WikiData[4] by their internal NER and entity linking tool. Then we can obtain the new representations of entities and clicked news for user modeling. The process can be calculated as follows:

$$G = \mathrm{GNN}([\mathrm{W}^e; \mathrm{D}], \mathrm{E}) \tag{5}$$

where W^e and D represent the entity and news set in clicked news, respectively. The entity set is made up of different entity embeddings $\mathbf{w}^e$. And the news set is composed of news $\mathbf{d}$ browsed by the user. E contains three types of edge, i.e., news-news, news-entity, entity-entity. For example, a user has three clicked news d_1, d_2, d_3. The entities in the news d_1 are *PGA TOUR*, *Joe Dey*. The entities in the news d_2 are *PGA TOUR*, *Korn Ferry Tour* and *Sky Sports*. The entities in the news d_3 are *United Kingdom*, *Sky Sports*. Thus, the nodes in this graph includes three news nodes d_1, d_2, d_3 and corresponding entity nodes for the user. The edges of news-news are $d_1 - d_2$ and $d_2 - d_3$, which also represents the chronological order of clicks. The edges of news-entity are d_1–*PGA TOUR*, d_1–*Joe Dey* and so on, which represents the affiliation relationship between entity and news. The edges of entity-entity are *PGA TOUR* (in d_1)-*PGA TOUR* (in d_2) and *Sky Sports* (in d_2)-*Sky Sports* (in d_3), which are bidirectional and connect the same entity in different news.

After we feed these nodes and edges to graph neural networks for reasoning, we can obtain the new representation of these nodes $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_P]$, where P represents the total number of clicked news and corresponding entities.

4.3 User Information Aggregation

Mentioned by ResNet [3], we combine original implicit news representation with explicit representation of news nodes in this part, i.e., $\mathbf{d}_i^g = [\mathbf{d}_i; \mathbf{d}_i']$, where $\mathbf{d}_i$ is original representation of the i-th news obtained from implicit text information, $\mathbf{d}_i'$ is new representation of the i-th news obtained from graph $\mathbf{G}$. Finally, we use similar attention function to aggregate clicked news for user representation $\mathbf{u}$, which can be calculated as follows:

$$\begin{aligned} c_i^g &= \mathbf{q}_g^T \tanh(\mathbf{V}_g \times \mathbf{d}_i^g + \mathbf{v}_g), \\ \alpha_i^g &= \frac{e^{c_i^g}}{\sum_{j=1}^{Q} e^{c_j^g}}, \\ \mathbf{u} &= \sum_{i=1}^{Q} \alpha_i^g \mathbf{d}_i^g, \end{aligned} \tag{6}$$

where $\mathbf{V}_g$ and $\mathbf{v}_g$ are the projection parameters, $\mathbf{q}_g$ denotes the attention query vector to be learned.

4.4 Prediction and Training

After the representation of a candidate news $\mathbf{d}_c$ and the representation of user $\mathbf{u}$ are obtained, the model can predict the probability

[4]https://www.wikidata.org/wiki/Wikidata:MainPage

of the user browsing candidate news based on their representations. Following [11], the click probability score $\hat{y}$ is calculated by the inner product of the representation vectors of user u and the candidate news d_c, i.e., $\hat{y} = \mathbf{u}^T \mathbf{d}_c$. According to previous research [11, 16], the inner product is not only the one with the best time efficiency but also the one with the best performance.

And during training, we use negative sampling techniques motivated by [6, 16, 20]. For each news browsed by a user which is regarded as a positive sample, we randomly sample K news which are presented in the same session but are not clicked by this user as negative samples. We then jointly predict the click probability scores of the positive news $\hat{y}^+$ and the K negative news $[\hat{y}_1^-, \hat{y}_2^-, ..., \hat{y}_K^-]$. In this way, we formulate the news click prediction problem as a pseudo $K + 1$-way classification task. We normalize these click probability scores using softmax to compute the posterior click probability of a positive sample. Then we use the negative log-likelihood of all positive samples as the loss function, which can be formulated as follows:

$$\mathcal{L} = -\sum_{i \in \mathcal{T}} \log\left(\frac{e^{\hat{y}_i^+}}{e^{\hat{y}_i^+} + \sum_{j=1}^{K} e^{\hat{y}_{i,j}^-}}\right) \tag{7}$$

where $\mathcal{T}$ is the set of the positive training samples, $\hat{y}_i^+$ is the click probability score of the i-th positive news, and $\hat{y}_{i,j}^-$ is the click probability score of the j-th negative news in the same session with the i-th positive news.

5 EXPERIMENTS

5.1 Datasets and Parameter Settings

We conduct our experiments on the MIcrosoft News Dataset (MIND) [19], a large-scale dataset for news recommendation research. MIND contains about 160k English news articles and more than 15 million impression logs generated by 1 million users, who had at least 5 news clicks during 6 weeks from October 12 to November 22, 2019. Every news article contains rich textual content including title, abstract, body, category and entities. Each impression log contains the click events, non-clicked events and historical news click behaviors of this user before this impression. To protect user privacy, each user was de-linked from the production system when securely hashed into an anonymized ID.

Following official split, we use the impression logs in the last week for test, and the logs in the fifth week for training. For samples in training set, the click behaviors in the first four weeks is used to construct the news click history for user modeling. For samples in test set, the time period for news click history extraction is the first five weeks. And we use 10% of the test data as our test set.

In our experiments, the dimensions of word embeddings are set to 300. We use the pre-trained Glove embedding [12] in our approach. The number of CNN filters is set to 300, and the window size is 3. The sizes of attention queries is 200. The negative sampling ratio K and batch size are set to 4 and 64, respectively. Adam [7] is used as the optimization algorithm. We apply 20% dropout [14] to each layer in our model to alleviate overfitting. These hyper-parameters are selected according to the validation set. AUC, MRR, nDCG@5 and nDCG@10 scores are adopted as our metrics.

Table 1: Performance comparison on the test set.

Models	AUC	MRR	nDCG@5	nDCG@10
NAML	0.6861	0.3405	0.3727	0.4299
NRMS	0.6864	0.3426	0.3751	0.4319
EEG (Single)	**0.6873**	**0.3435**	**0.3762**	**0.4329**
Softmax Ensemble	0.6966	0.3472	0.3811	0.4380
Scaled Ensemble	**0.6974**	**0.3475**	**0.3814**	**0.4383**

Table 2: Performance analysis on the validation set.

Methods	AUC	MRR	nDCG@5	nDCG@10
EEG	0.6861	0.3284	0.3645	0.4295
EEG-concat	0.6851	0.3298	0.3642	0.4292
EEG-affiliation	0.6788	0.3205	0.3555	0.4216
EEG-entity	0.6603	0.3103	0.3433	0.4089

5.2 Results and Analysis

We evaluate the performance of our model (EEG) with some strong baseline methods, including NAML [16] and NRMS [18]. We first conduct experiments on the test set in Table 1. Our single model performs better than existing methods. And we also apply some ensemble techniques to these three models. Softmax ensemble means using the average of softmax logits as the score. Scaled ensemble means using the average of normalized prediction value as the score. Experiments show that scaled ensemble methods achieves better scores than softmax ensemble.

We also conduct experiments to compare different strategies of our approach on the validation set. The results are shown in Table 2. EEG-concat changes the order in user information aggregation. We first aggregate news representation in news extractor and entity graph by attention mechanism, separately. Then we concentrate them as user representation. EEG-affiliation ignores interaction edges among different news and only keep the edges of news-entity. EEG-entity replaces aggregated news representation in GNN with aggregated initial entity representation. From these experiments, we can find that explicit entity graph can connect different news effectively for user modeling. And the performance may drop a lot if we simply introduce entities into user representations.

6 CONCLUSION

In this paper, we propose a novel news recommendation approach, EEG, which combines the explicit entity graph with implicit text information. It can enhance user representation sufficiently and precisely. The entity graph consists of two types of nodes and three kinds of edges. The nodes can represent news and entities. And the edges can represent chronological order, related and affiliation relationship. Then graph neural network is utilized for reasoning on this graph. Extensive experiments on a real-world dataset, Microsoft News Dataset (MIND), validate the effectiveness of our proposed approach. In our future work, we will further try different applications of graph neural networks in news recommendation.

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*. Association for Computational Linguistics, Florence, Italy.

[2] Trapit Bansal, Mrinal Das, and Chiranjib Bhattacharyya. 2015. Content driven user profiling for comment-worthy recommendations of news and blog articles. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 195–202.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *European conference on computer vision*. Springer, 630–645.

[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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online.

[6] 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 *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.

[7] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[8] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2018. Towards Better Representation Learning for Personalized News Recommendation: a Multi-Channel Deep Fusion Approach.. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. 3805–3811.

[9] 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 *Fourteenth ACM Conference on Recommender Systems*. 200–209.

[10] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. 31–40.

[11] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-Based News Recommendation for Millions of Users. Association for Computing Machinery.

[12] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*. 1532–1543.

[13] Jeong-Woo Son, A-Yeong Kim, and Seong-Bae Park. 2013. A location-based news article recommendation with explicit localized semantic analysis. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 293–302.

[14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[15] 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.

[16] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Attentive Multi-View Learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 3863–3869.

[17] 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.

[18] 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*. Association for Computational Linguistics, Hong Kong, China.

[19] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A Large-scale Dataset for News Recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online.

[20] Shuangfei Zhai, Keng-hao Chang, Ruofei Zhang, and Zhongfei Mark Zhang. 2016. Deepintent: Learning attentions for online advertising with recurrent neural networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1295–1304.

[21] Xuanyu Zhang. 2020. CFGNN: Cross Flow Graph Neural Networks for Question Answering on Complex Tables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9596–9603.