User-as-Graph: User Modeling with Heterogeneous Graph Pooling for News Recommendation

Chuhan Wu[1], **Fangzhao Wu**[2], **Yongfeng Huang**[1], **Xing Xie**[2]
[1]Department of Electronic Engineering & BNRist, Tsinghua University, Beijing 100084, China
[2]Microsoft Research Asia, Beijing 100080, China
{wuchuhan15, wufangzhao}@gmail.com, yfhuang@tsinghua.edu.cn, xing.xie@microsoft.com

Abstract

Accurate user modeling is critical for news recommendation. Existing news recommendation methods usually model users' interest from their behaviors via sequential or attentive models. However, they cannot model the rich relatedness between user behaviors, which can provide useful contexts of these behaviors for user interest modeling. In this paper, we propose a novel user modeling approach for news recommendation, which models each user as a personalized heterogeneous graph built from user behaviors to better capture the fine-grained behavior relatedness. In addition, in order to learn user interest embedding from the personalized heterogeneous graph, we propose a novel heterogeneous graph pooling method, which can summarize both node features and graph topology, and be aware of the varied characteristics of different types of nodes. Experiments on large-scale benchmark dataset show the proposed methods can effectively improve the performance of user modeling for news recommendation.

1 Introduction

News recommendation techniques are adopted by many online news platforms to provide personalized news services and alleviate information overload of users [Okura *et al.*, 2017]. Precise user interest modeling is a prerequisite for accurate personalized news recommendation. Existing methods for news recommendation mainly model users' interest from their clicked news via sequential or attentive models. For example, Okura et al. [2017] proposed to use a GRU network to learn user representations from clicked news by capturing their sequential information. Wu et al. [2019a] proposed to use attention network to learn user representations from clicked news by attending to important news. However, these methods cannot effectively model the relatedness between user behaviors, which can usually provide useful behavior contexts for modeling user interest.

In fact, user behaviors on news may have multiple kinds of relatedness. First, the clicked news that are adjacent in time may have some relatedness. For example, as shown in Fig. 1, the user clicks the first and second news that both mention

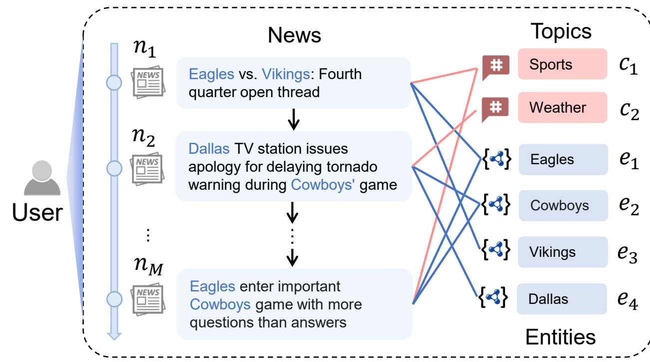

Figure 1: Personalized heterogeneous graph of an example user.

news events on NFL. In addition, the news belonging to the same topic category may also have some relatedness. For example, the first and last news in Fig. 1 are in the "sports" category and they are similar in content. Besides, the news containing the same entities also have some relations in attracting news clicks, because entities in news are important indications of user interest [Wang *et al.*, 2018]. For instance, the second and last news in Fig. 1 both mention the entity "Cowboys", which is a football team name. We can infer that this user clicks both news due to his/her potential interest in this team. These kinds of relations between user behaviors can provide very useful clues to model the contexts of behaviors for more accurate user interest modeling. However, in existing news recommendation methods that mainly represent users with their behavior sets or sequences, the rich relatedness between user behaviors is not effectively modeled, which may be not optimal.

In order to handle this problem, in this paper we propose a user modeling method named *User-as-Graph (UaG)* for news recommendation, which represents each user as a personalized heterogeneous graph built from their behaviors to better capture the fine-grained behavior relatedness for user modeling. The user graph nodes contains the clicked news, topics and entities, and edges are created between a user's adjacent clicked news, news and its topic, as well as news and its entities. Different from existing methods that can use sequential or attentive models to process the behavior sequences or sets, it is difficult to learn user embeddings from the personalized heterogeneous graph. Thus, we propose a heterogeneous graph pooling *(HG-Pool)* method to iteratively condense the personalized heterogeneous graph and meanwhile consider the different characteristics of different types of nodes. It uses different pooling graph neural networks for different kinds of nodes to summarize type-specific graph information from the entire graph, and aggregates the same kinds of nodes based on their features and the entire graph topology. Extensive experiments on large-scale benchmark dataset show that our approach can effectively enhance the performance of user modeling for news recommendation.

The major contributions of this paper include:

- We propose a novel user modeling method which represents each user as a personalized heterogeneous graph to better model the relatedness between user behaviors for more accurate user interest modeling.
- We propose a novel heterogeneous graph pooling method to learn user interest embedding from the personalized heterogeneous graphs. To our best knowledge, it is the first work on heterogeneous graph pooling.
- We conduct extensive experiments on large-scale dataset to verify the effectiveness of the proposed methods.

2 Related Work

2.1 User Modeling for News Recommendation

User interest modeling is a core problem in news recommendation. Many existing methods model users' interests from their clicked news [Wu *et al.*, 2019b; Wang *et al.*, 2020; Lee *et al.*, 2020; Wu *et al.*, 2020a; Wu *et al.*, 2020b; Wu *et al.*, 2021]. For example, Okura et al. [2017] proposed to use a GRU network to learn user embeddings from clicked news embeddings. Wang et al. [2018] proposed to learn representations of users from their clicked news using a candidate-aware attention network that evaluates their relevance to the candidate news. Wu et al. [2019c] proposed to use multi-head self-attention to capture the interactions between clicked news and learn user representations via attention pooling. These methods usually represent users by their behavior set or sequence, and learn user representations via attentive or sequential models. The rich relatedness between user behaviors cannot be effectively captured by these methods, which may be suboptimal in modeling user interest. There are a few news recommendation methods that exploit high-order information on graphs for user modeling [Hu *et al.*, 2020; Ge *et al.*, 2020]. For example, Hu et al. [2020] proposed to use a graph neural network to learn user interest embeddings from a user-news-topic graph. Ge et al. [2020] proposed to use neighbors of users on the user-news graph to enhance user representations. In these methods, each user is only represented by a node in a global user-news graph. Differently, in our approach we represent each user with a personalized graph built from user behaviors, which can provide finer-grained information for inferring user interests.

2.2 Graph Pooling

Graph pooling aims to condense an input graph into a smaller one for efficient representation learning [Ying *et al.*, 2018]. However, different from max and attentive pooling that are widely used for processing other data genres like images and texts, graph pooling is non-trivial because it is difficult to decide which nodes to retain according to both their own features and the graph topology [Lee *et al.*, 2019]. Several early graph pooling methods usually pool a graph globally with a single layer. For example, Duvenaud et al. [2015] proposed to sum up or average all node embeddings to form the graph representation. Li et al. [2016] proposed to introduce a "super node" that has connections with each node in the graph, and use the embedding of this node as the graph representation. However, these methods usually cannot effectively exploit the graph topology, which is usually critical for learning accurate graph representations.

In recent years, several works explore to pool graph data in a hierarchical way [Yuan and Ji, 2019; Yang *et al.*, 2020; Ranjan *et al.*, 2020]. For example, Ying et al. [2018] proposed a differentiable pooling method named DiffPool. It first learns soft clusters for nodes in a GNN layer, and then aggregates the nodes in each cluster to form a coarsened graph representation as the input of the next GNN layer. Gao and Ji [2019] proposed a graph pooling method named gPool. After each GCN layer, they used a query vector to compute a score for each node, and only retained the top K nodes to form a concentrated graph. The scores of the retained nodes were multiplied with their node embeddings to form new node features. Lee et al. [2019] proposed a self-attentive graph pooling (SAGPool) method. Different from gPool, they used a graph neural network instead of a query vector to capture the graph topology. These graph pooling methods are mainly designed for homogeneous graphs where nodes are not distinguished by their types. However, in our *User-as-Graph* approach, the personalized graph for each user is a heterogeneous graph. In heterogeneous graphs different kinds of nodes may have differences in their node features and topology, and it may not be optimal to directly apply these methods to heterogeneous graphs. However, to our best knowledge, heterogeneous graph pooling is rarely studied. Thus, we propose a heterogeneous graph pooling method that can effectively summarize the topological and content information of heterogeneous graphs and be aware of the differences between different kinds of nodes.

3 User-as-Graph for User Modeling

We introduce our *User-as-Graph (UaG)* approach for user modeling in news recommendation. In *User-as-Graph*, each user is represented as a personalized heterogeneous graph constructed from their behaviors. To learn user interest embedding for news recommendation from the personalized heterogeneous graph, we propose a novel heterogeneous graph pooling method to iteratively condense this graph. The details of our approach are introduced as follows.

3.1 Personalized Graph Construction

First, we introduce the personalized graph in our *User-as-Graph* method for user interest modeling, which is constructed from the news click behaviors of a user. It contains three kinds of nodes, including news, topic and entities.

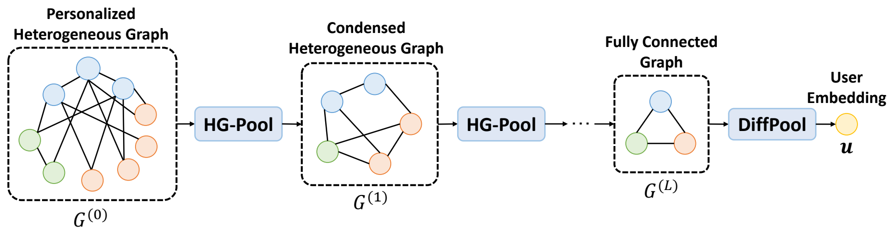

Figure 2: User representation learning is formulated as heterogeneous graph pooling in our *User-as-Graph* method.

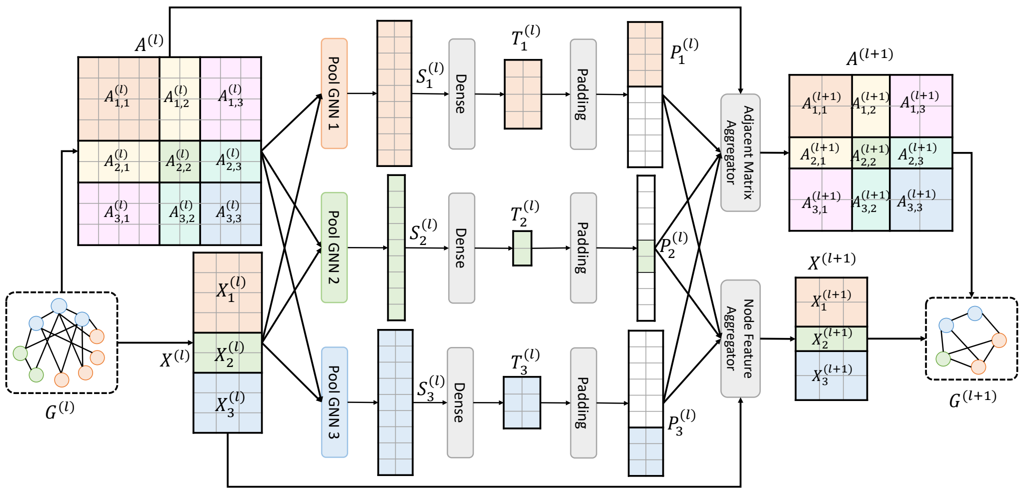

Figure 3: The proposed *HG-Pool* method for heterogeneous graph pooling.

Given a target user u, we denote the set of M news clicked by this user as $\mathcal{N} = \{n_1, n_2, ..., n_M\}$, which are ordered by their click time. User behaviors that are adjacent in time usually have some relatedness. For example, in Fig. 1 the user successively clicks two news about similar events. The sequential information of news clicks is important for modeling the interest dynamics of users. Thus, we regard each clicked news as a node and further link the news adjacent in time (i.e., n_i with n_{i+1}, $1 \le i < M$). In addition, news topics are very useful clues for modeling news content and user interest. The news with the same topic category may also have some relations in user interest modeling. Thus, we incorporate topic information into the user graph by regarding each topic category c_i as a node, and connect each news node n_i with its associated topic node $c_{[n_i]}$. Besides, news entities are very important for news and user modeling because users may often decide to click a news due to the attraction of the entities mentioned in news titles, such as famous persons or organizations like sports teams. To incorporate entity information, we regard entities as another kind of nodes and integrate the set of entities $\mathcal{E} = \{e_1, e_2, ..., e_K\}$ (K is the entity number) that are mentioned by the news in $\mathcal{N}$ into the user graph. We connect each news node n_i with the nodes of its mentioned entities to finalize the personalized heterogeneous graph $G^{(0)}$.

3.2 Heterogeneous Graph Pooling

Since in our *User-as-Graph* method each user is represented as a personalized heterogeneous graph $G^{(0)}$, it is essential to learn a user interest embedding $\mathbf{u}$ from it for news recommendation. A natural way for this goal is graph pooling. Existing graph pooling methods are mostly designed for homogeneous graphs where nodes have similar properties. However, in heterogeneous graphs different kinds of nodes have different features and topology. Thus, it is not optimal to directly apply existing homogeneous graph pooling methods to the personalized heterogeneous graphs. To handle this problem, we propose a heterogeneous graph pooling (HG-Pool) method that can consider the varied characteristics of different kinds of nodes for graph representation learning. Its architecture is shown in Fig. 3.

For a heterogeneous graph with T types of nodes (T is 3 for $G^{(0)}$), we respectively denote its initial graph adjacent matrix and node features as $A^{(0)}$ and $X^{(0)}$. We apply multiple GNN layers[1] to process the graph based on the input $A^{(0)}$ and $X^{(0)}$, and the output by the l-th GNN layer as $A^{(l)} \in \mathbb{R}^{N^{(l)} \times N^{(l)}}$ and $X^{(l)} \in \mathbb{R}^{N^{(l)} \times D}$, where $N^{(l)} = \sum_{i=1}^{T} N_i^{(l)}$ is the summation of the number $N_i^{(l)}$ of each kind of node, and D is the node feature dimension. Since different kinds of nodes may have diverse characteristics, we need to distinguish the nodes in a heterogeneous graph by types. Specifically, we divide the graph adjacent matrix $A^{(l)}$ into T^2 sub-matrices and the node feature matrix $X^{(l)}$ into T sub-matrices according to node types. We denote the adjacent sub-matrix in the i-th

[1] Can be implemented by various methods like GCN and GAT.

row and the j-th column as $A_{i,j}^{(l)} \in \mathbb{R}^{N_i^{(l)} \times N_j^{(l)}}$ (representing the connections between the i-th and the j-th types of nodes), and the i-th feature sub-matrix as $X_i^{(l)} \in \mathbb{R}^{D \times N_i^{(l)}}$ (representing the features of the i-th type of node). Motivated by DiffPool [Ying *et al.*, 2018], we propose to apply T pooling graph neural networks to independently learn a pooling matrix for each kind of node as follows:

$$S_i^{(l)} = \text{PoolGNN}(A^{(l)}, X^{(l)}; \Theta_i^{(l)}), \tag{1}$$

where $S_i^{(l)} \in \mathbb{R}^{N^{(l)} \times N_i^{(l+1)}}$ is the pooling matrix for the i-th node type, and $\Theta_i^{(l)}$ is the parameter set of this pooling GNN. In this way, the characteristics of different kinds of nodes can be modeled by different node-specific pooling matrix $S_i^{(l)}$.

In existing homogeneous graph pooling methods like DiffPool, the summarized matrix output by the GNN pooling will be applied to the adjacent matrix $A^{(l)}$ and the node feature matrix $X^{(l)}$. However, it will cluster different kinds of nodes together without discrimination, which is not suitable for heterogeneous graphs pooling since their heterogeneous properties cannot be retained. Thus, we propose to condense the pooling matrix $S_i^{(l)}$ into a summarized one $T_i^{(l)} \in \mathbb{R}^{N_i^{(l)} \times N_i^{(l+1)}}$ that conveys node-specific graph information, which is formulated as:

$$T_i^{(l)} = \text{softmax}(W_i^{(l)} S_i^{(l)} + B_i^{(l)}), \tag{2}$$

where $W_i^{(l)}$ and $B_i^{(l)}$ are parameters for summarizing $S_i^{(l)}$. To avoid indexing operations, we apply zero paddings[2] to $T_i^{(l)}$ to obtain an aligned pooling matrix $P_i^{(l)} \in \mathbb{R}^{N^{(l)} \times N_i^{(l+1)}}$.

Then, we propose an adjacent matrix aggregator and a node feature aggregator to compute the new adjacent matrix $A^{(l+1)}$ and node feature $X^{(l+1)}$. The adjacent matrix aggregator takes the previous adjacent matrix $A^{(l)}$ and the T aligned pooling matrices $[P_1^{(l)}, ..., P_T^{(l)}]$ as the input. We denote the new adjacent sub-matrix in the i-th row and the j-th column as $A_{i,j}^{(l+1)} \in \mathbb{R}^{N_i^{(l+1)} \times N_j^{(l+1)}}$, which is computed as:

$$A_{i,j}^{(l+1)} = P_i^{(l)\top} A^{(l)} P_j^{(l)}. \tag{3}$$

The entire pooled adjacent matrix $A^{(l+1)} \in \mathbb{R}^{N^{(l+1)} \times N^{(l+1)}}$ is the 2-D concatenation of all the adjacent sub-matrices. In this way, different types of nodes will not be clustered together and the properties of heterogeneous graph can be retained. In addition, the relations among different kinds of nodes can also be modeled using the interaction between different pooling matrices. The node feature aggregator receives the node feature matrix $X^{(l)}$ and the aligned pooling matrices as input. We denote the new feature sub-matrix of the i-th kind of nodes as $X_i^{(l+1)}$, which is formulated as:

$$X_i^{(l+1)} = P_i^{(l)\top} X^{(l)}. \tag{4}$$

The output node feature matrix is the concatenation of the feature sub-matrices of the T kinds of nodes, i.e., $X^{(l+1)} = concat(X_1^{(l+1)}, X_2^{(l+1)}, ..., X_T^{(l+1)})$.

[2]Only rows corresponding to the i-th kind of nodes are non-zero.

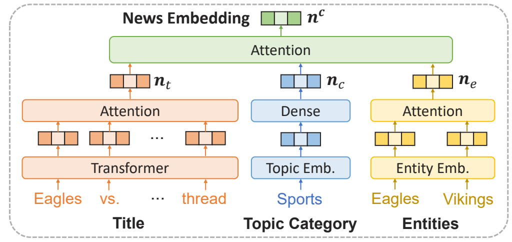

Figure 4: The architecture of the news encoder.

In the heterogeneous graph pooling module, we apply our *HG-Pool* method to each GNN layer and use multiple stacks of them to learn graph representations in a hierarchical manner, as shown in Fig. 2. After applying *HG-Pool* iteratively for L rounds, the original heterogeneous graph will be condensed into a small densely connected graph with T nodes, where each node condenses the information of its corresponding type of nodes in the original graph. Finally, we apply a DiffPool [Ying *et al.*, 2018] layer to convert this graph into a single node that summarizes the information of the entire graph. The embedding of this node is output as the user interest embedding $\mathbf{u}$ for news recommendation.

3.3 Node Representation Learning

Next we introduce how to learn initial node representations in the personalized heterogeneous graph. To learn news node representations, motivated by [Wu *et al.*, 2019c] we use a Transformer to learn word representations from news title, and use an attention network to form the news node representations by attending to important words. To represent topic nodes, we use the embeddings of topic categories as the node representations. For the entities nodes, following [Wang *et al.*, 2018] we first learn entity embeddings from the WikiData knowledge graph via the TransE [Bordes *et al.*, 2011] method, and apply a dense layer to these embeddings to align the feature dimensions. Note that entity embeddings are fine-tuned during model training.

3.4 Model Training

We train the User-as-Graph model in a news click prediction task. We use a news encoder to learn candidate news embeddings and predict the click scores of them for model training based on their relevance to the user interest embedding learned by our User-as-Graph model. More specifically, in the news encoder, we use the attentive multi-view learning framework [Wu *et al.*, 2019a] to incorporate news title, topic category and entities as different views of a news, as shown in Fig. 4. Similar to node representation learning, we use a Transformer and attention network to learn the title representation $\mathbf{n}_t$, and learn the hidden topic representation $\mathbf{n}_c$ by applying an additional dense layer after the topic embedding layer. In addition, we apply an attention network after the entity embedding layer to learn the entity-based news representation $\mathbf{n}_e$ by selecting important entities. The final unified candidate news embedding $\mathbf{n}^c$ is aggregated from the three kinds of news representations via an attention network. The click probability score y of the user u clicking the candidate news n^c is computed by the inner product between their embeddings, i.e., $y = \mathbf{u}^\top \mathbf{n}^c$. The scores are used for personalized news ranking and display. Following [Wu *et al.*, 2019c], for each clicked news we randomly select P non-clicked news that are displayed in the same impression to build training samples. The loss function for model training is formulated as follows:

$$\mathcal{L} = -\frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\log\left(\frac{\exp(y_i)}{\exp(y_i)+\sum_{j=1}^{P}\exp(y_{i,j})}\right), \tag{5}$$

where $\mathcal{S}$ is the training set, y_i and $y_{i,j}$ denote the predicted click score of the i-th clicked sample and its associated j-th non-clicked sample respectively.

4 Experiments

4.1 Datasets and Experimental Settings

Our experiments are conducted on a large-scale public news recommendation dataset named MIND [Wu *et al.*, 2020c][3], which contains the news impression logs of 1 million users from Microsoft News[4] in 6 weeks (from Oct. 12 to Nov. 22, 2019). The samples in the last week are reserved for test, and those in the first 5 weeks are used for training and validation. The detailed statistics of this dataset are shown in Table 1.

# Users	1,000,000	Avg. # words per news title	11.52
# News	161,013	# Click behaviors	24,155,470
# Impressions	15,777,377	# Topic categories	20
# Entities	3,299,687	Avg. # entities per news	16.71

Table 1: Detailed dataset statistics.

In our experiments, we use GAT [Veličković *et al.*, 2018] to implement the graph neural networks. We use Glove [Pennington *et al.*, 2014] to initialized word embeddings. The number of *HG-Pool* layers is 2. Adam [Kingma and Ba, 2015] is used as the optimizer. The hyperparameters are tuned on the validation set. Each experiment is repeated 5 times. Following [Wu *et al.*, 2019c], we report the average AUC, MRR, nDCG@5 and nDCG@10 scores over all impressions.

4.2 Performance Evaluation

First, we compare our *User-as-Graph* method with many baselines, including: (1) *LibFM* [Rendle, 2012], a popular FM tool for recommendation. (2) *EBNR* [Okura *et al.*, 2017], embedding-based news recommendation with autoencoders and GRU network. (3) *DKN* [Wang *et al.*, 2018], deep knowledge-aware network for news recommendation. (4) *NPA* [Wu *et al.*, 2019b], news recommendation with personalized attention. (5) *NAML* [Wu *et al.*, 2019a], news recommendation with attentive multi-view learning. (6) *LSTUR* [Wu *et al.*, 2019a], news recommendation with long- and short-term user representations. (7) *NRMS* [Wu *et al.*, 2019c], a news recommendation method based on multi-head self-attention. (8) *GNewsRec* [Hu *et al.*, 2020], using attentive LSTM to model short-term user interest and GNN to model long-term user interest from a user-news-topic graph. (9) *GERL* [Ge *et al.*, 2020], using the neighbors of news and users on the user-news graph to enhance their representations.

[3]https://msnews.github.io/

[4]https://www.msn.com/en-us

Methods	AUC	MRR	nDCG@5	nDCG@10
LibFM	59.93	28.23	30.05	35.74
EBNR	65.42	31.24	33.76	39.47
DKN	64.60	31.32	33.84	39.48
NPA	66.69	32.24	34.98	40.68
NAML	66.86	32.49	35.24	40.91
LSTUR	67.73	32.77	35.59	41.34
NRMS	67.76	33.05	35.94	41.63
GNewsRec	67.53	32.68	35.46	41.17
GERL	68.24	33.46	36.38	42.11
User-as-Graph	**69.23**	**34.14**	**37.21**	**43.04**

Table 2: Performance of different methods. Improvement over the second best results is significant at $p < 0.05$.

	AUC	MRR	nDCG@5	nDCG@10
News	68.10	33.37	36.30	42.01
News+Topic	68.69	33.84	36.77	42.53
News+Entity	68.78	33.90	36.85	42.61
News+Topic+Entity	69.23	34.14	37.21	43.04

Table 3: Effects of different kinds of nodes for *User-as-Graph*.

The results are shown in Table 2, which reveal several findings. First, deep learning-based news recommendation methods (e.g., *EBNR*, *NRMS* and *User-as-Graph*) outperform the traditional FM-based method that uses handcrafted features to represent news and users. It shows that neural models are better at modeling news content and user interest than handcrafted features. Second, the methods that consider the relatedness between clicked news (e.g., *NRMS*, *GNewsRec* and *User-as-Graph*) usually outperform the methods that ignore. This is probably because capturing the relations between clicked news can help model user interest. Third, our *User-as-Graph* method consistently outperforms other baseline methods, and its advantage over the best performed one is significant. This is because *User-as-Graph* models user interests with a personalized graph, which can model the fine-grained relatedness between user behaviors. In addition, our approach learns user embeddings via heterogeneous graph pooling, which can capture the high-order information on the personalized graph and meanwhile fully consider the characteristics of heterogeneous graphs. Thus, our approach outperforms the baseline methods.

4.3 Ablation Study

Next, we verify the effectiveness of different kinds of nodes in the personalized graph, i.e., news titles, topics and entities. The results are shown in Table 3. We find that both topics and entities nodes can improve the model performance. This is probably because news topics can provide a coarse-grained understanding of news content, and entities usually condense the key elements of news and can help capture the fine-grained user interests on news entities. Besides, combining all three kinds of nodes can further improve the model performance, which verifies their effectiveness.

4.4 Model Effectiveness

Next, we verify the effectiveness of *User-as-Graph* in user modeling by comparing it with several other user modeling

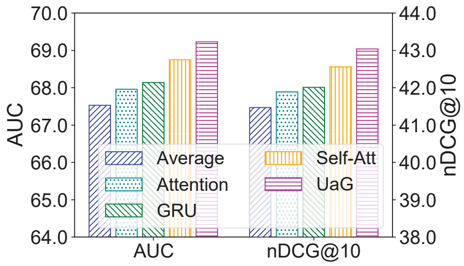

Figure 5: Effect of *User-as-Graph (UaG)* for user modeling.

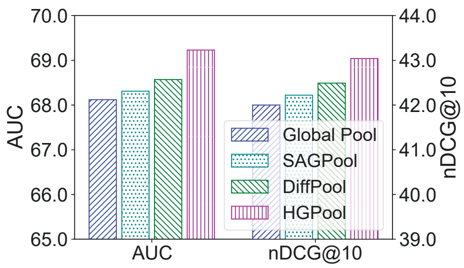

Figure 6: Effect of *HG-Pool* for heterogeneous graph pooling.

methods, including: averaging clicked news embeddings (*Average*), attention network (*Attention*), GRU network (*GRU*), and self-attention network (*Self-Att*). For fair comparison, all methods use titles, topics and entities. The results are shown in Fig. 5. We find that *Average* is inferior to other methods. This is because it cannot distinguish the importance of different clicked news for user modeling. In addition, *GRU*, *Self-Att* and *User-as-Graph* outperform *Attention*. This is because they can model relatedness between clicked news which is ignored in attention network. Besides, *User-as-Graph* outperforms other user modeling methods like *GRU* and *Self-Att*. This is because our *User-as-Graph* method can model the various relationship among clicked news, topics, and entities on the personalized graphs, while other methods cannot. It verifies the effectiveness of modeling user interests with personalized graphs.

We also conducted experiments to show the advantage of our proposed heterogeneous graph pooling method (*HG-Pool*) over existing graph pooling methods in user interest modeling. We compare our method with global pooling [Duvenaud *et al.*, 2015], SAGPool [Lee *et al.*, 2019] and DiffPool [Ying *et al.*, 2018]. The results are shown in Fig. 6. We find that hierarchical graph pooling methods are better than global graph pooling. This may be because global graph pooling cannot capture the hierarchical graph structure. In addition, we find that the performance of *SAGPool* is also not optimal. This may be because most nodes in our user graph can provide useful clues for understanding user interests, and it may not be optimal to directly discard some of them. Besides, our *HG-Pool* method outperforms other compared graph pooling methods. This is because *HG-Pool* can consider the different characteristics of different kinds of nodes while other methods cannot.

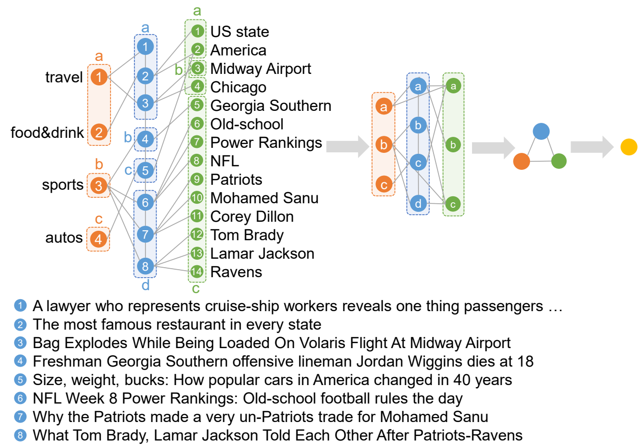

Figure 7: The iterative pooling process of the personalized heterogeneous graph constructed from user behaviors. Blue, orange, and green nodes stand for news, topics and entities, respectively.

4.5 Case Study

Finally, we conduct several case studies to show how our proposed *User-as-Graph* method works. The personalized graph of a randomly selected user and its pooling process are shown in Fig. 7. From the results, we find that our method can effectively cluster homogeneous nodes that have inherent relatedness. For example, the news (6)-(8) are clustered together because they share the same topic and are adjacent in time. The nodes of the topic "travel" and "food&drink" are aggregated together because from the news they connected, we can infer that this user may be interested in finding nice cuisine in travel. In addition, we find that our model can also create useful node connections. For example, although there is no direct connection between entity and topic nodes, our model links the topic cluster *b* to the entity cluster *c*, which indicates the interactions between the entities about sports and the "sports" topic category. These results show the effectiveness of *User-as-Graph* for user interest modeling from the personalized heterogeneous graph via *HG-Pool*.

5 Conclusion

In this paper, we propose a novel user modeling method for news recommendation, which represents each user as a personalized heterogeneous graph built from their behaviors. It can capture the rich relatedness between user behaviors to enhance user interest modeling. We also propose a heterogeneous graph pooling method to learn user interest representation from the personalized heterogeneous graph. Experimental results on a large-scale benchmark dataset show that our User-as-Graph method can effectively improve the performance of user modeling for news recommendation.

Acknowledgments

This work is supported the National Natural Science Foundation of China (NSFC) under Grants U1936208, U1936216, U1836204 and U1705261.

References

[Bordes *et al.*, 2011] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *AAAI*, pages 301–306, 2011.

[Duvenaud *et al.*, 2015] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NIPS*, pages 2224–2232, 2015.

[Gao and Ji, 2019] Hongyang Gao and Shuiwang Ji. Graph u-nets. In *ICML*, pages 2083–2092, 2019.

[Ge *et al.*, 2020] Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. Graph enhanced representation learning for news recommendation. In *WWW*, pages 2863–2869, 2020.

[Hu *et al.*, 2020] Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. Graph neural news recommendation with long-term and short-term interest modeling. *Inf. Process. & Manage.*, 57(2):102142, 2020.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Lee *et al.*, 2019] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *ICML*, pages 6661–6670, 2019.

[Lee *et al.*, 2020] Dongho Lee, Byungkook Oh, Seungmin Seo, and Kyong-Ho Lee. News recommendation with topic-enriched knowledge graphs. In *CIKM*, pages 695–704, 2020.

[Li *et al.*, 2016] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *ICLR*, 2016.

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

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Ranjan *et al.*, 2020] Ekagra Ranjan, Soumya Sanyal, and Partha P Talukdar. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In *AAAI*, pages 5470–5477, 2020.

[Rendle, 2012] Steffen Rendle. Factorization machines with libfm. *TIST*, 3(3):57, 2012.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Wang *et al.*, 2018] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation. In *WWW*, pages 1835–1844, 2018.

[Wang *et al.*, 2020] Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. Fine-grained interest matching for neural news recommendation. In *ACL*, pages 836–845, 2020.

[Wu *et al.*, 2019a] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. Neural news recommendation with attentive multi-view learning. In *IJCAI*, pages 3863–3869, 2019.

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

[Wu *et al.*, 2019c] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. Neural news recommendation with multi-head self-attention. In *EMNLP-IJCNLP*, pages 6390–6395, 2019.

[Wu *et al.*, 2020a] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. Sentirec: Sentiment diversity-aware neural news recommendation. In *AACL-IJCNLP*, pages 44–53, 2020.

[Wu *et al.*, 2020b] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. User modeling with click preference and reading satisfaction for news recommendation. In *IJCAI*, pages 3023–3029, 2020.

[Wu *et al.*, 2020c] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. Mind: A large-scale dataset for news recommendation. In *ACL*, pages 3597–3606, 2020.

[Wu *et al.*, 2021] Chuhan Wu, Fangzhao Wu, Xiting Wang, Yongfeng Huang, and Xing Xie. Fairrec:fairness-aware news recommendation with decomposed adversarial learning. In *AAAI*, 2021.

[Yang *et al.*, 2020] Jia-Qi Yang, De-Chuan Zhan, and Xin-Chun Li. Bottom-up and top-down graph pooling. In *PAKDD*, pages 568–579. Springer, 2020.

[Ying *et al.*, 2018] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *NIPS*, pages 4800–4810, 2018.

[Yuan and Ji, 2019] Hao Yuan and Shuiwang Ji. Structpool: Structured graph pooling via conditional random fields. In *ICLR*, 2019.