The Journal of Supercomputing (2024) 80:14470–14488
https://doi.org/10.1007/s11227-024-06025-9

# Graph neural news recommendation based on multi-view representation learning

**Xiaohong Li[1] · Ruihong Li[1] · Qixuan Peng[1] · Jin Yao[1]**

Accepted: 24 February 2024 / Published online: 20 March 2024


**Abstract**
Accurate news representation is of crucial importance in personalized news recommendation. Most of existing news recommendation model lack comprehensiveness because they do not consider the higher-order structure between user–news interactions, relevance between user clicks on news. In this paper, we propose graph neural news recommendation based on multi-view representation learning which encodes high-order connections into the representation of news through information propagation along the graph. For news representations, we learn click news and candidate news content information embedding from various news attributes. And then combine obtained structure-based representations with representations from news content. Besides, we adopt a candidate-aware attention network to weight clicked news based on their relevance with candidate news to learn candidate-aware user interest representation for better matching with candidate news. The performance of the model has been improved in common evaluation metric. Extensive experiments on benchmark datasets show that our approach can effectively improve performance in news recommendation.

**Keywords** Graph neural network · Multi-head self-attention · User modeling · News recommendation · News modeling

Xiaohong Li and Ruihong Li have contributed equally to this work.

✉ Xiaohong Li
xiaohongli@nwnu.edu.cn

Ruihong Li
2021212111@nwnu.edu.cn

Qixuan Peng
2021222181@nwnu.edu.cn

Jin Yao
2022222228@nwnu.edu.cn

[1] College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, People's Republic of China

## 1 Introduction

With the rapid development of information technology, online news platforms such as Google News and Microsoft News have attracted a large number of users. However, a large number of news articles are created and published every day, and users cannot browse all valuable news. Users are flooded with news on various topics from around the world. In order to alleviate the problem of information overload, help users target their reading interests, and propose personalized news recommendation [1–3]. In recent years, news recommendation has become a hot research topic for many research institutions at home and abroad, and has received high attention from domestic and international academic conferences in the fields of information retrieval, data mining and artificial intelligence [4, 5].

Despite its significant success in improving the user experience, there are still many issues and challenges that require further research. Traditional news recommendation methods can be divided into content-based, collaborative filtering and hybrid. Early collaborative filtering (CF) algorithms [6] often had data sparsity and cold start issues. The content-based approach [7] is another type that extracts semantic and contextual information to mitigate the problems encountered by CF methods. The hybrid approach [8] tries to combine the two and achieve better performance. These methods describe the user's reading history and content, mainly studying how to effectively encode news and users into distinct representations. Although these methods have achieved good performance, there are still shortcomings in user interest representation and news embedding representation.

In recent years, many methods based on deep learning (DL) technology have been used for news modeling and have achieved good performance. In news recommendation, DL technology can extract deep level features contained in user interests and news content [9], and integrate multi-source heterogeneous auxiliary information that contains rich user interests, learning the unified representation of different data [10]. In addition, DL technology can alleviate issues such as cold start and data sparsity in news recommendation, effectively improving news recommendation performance. Most of which are attributed to content-based methods [11–13]. However, there are still many problems. Some of methods often lack comprehensiveness because they do not consider the higher-order structure between user–news interactions and the importance of candidate news for modeling user representations. We propose a graph neural news recommendation method based on multi-view representation learning (GRMR). In our method, users and news are both viewed as nodes in a bipartite graph constructed from historical user click behaviors. GRMR encodes higher-order information by spreading and embedding on the graph. This allows node information and topology to be integrated naturally, and dependencies between nodes can be effectively modeled. For news representations, we learn click news and candidate news content information embedding. Then we combine it with the information from high-order structure in the graph via a graph neural network. Finally, we use candidate-aware news representation to obtain user representation. Therefore, this article has the following three contributions.

- We propose a graph neural news recommendation method based on multi-view representation learning (GRMR), which encodes higher-order information by spreading and embedding on the graph.
- We construct a heterogeneous user–news graph to model user–item interactions, which alleviates the sparsity of user–item interactions. Then it applies graph neural networks to learn user and news embeddings with high-order information encoded by propagating embeddings over the graph.
- Experimental results on real-world datasets demonstrate the effectiveness of our method.

## 2 Related work

In news recommendation, many methods based on deep learning (DL) technology have been used for news modeling and have achieved good performance. These technology can integrate multi-source heterogeneous auxiliary information and extract deep level features of news content, then learning the unified representation of different data. Most of which are attributed to content-based methods. For example, NPA [14] is a neural news recommendation model based on personalized attention, which uses the embedding of user IDs to generate query vectors for words and news level attention, thereby selecting important news and important words in the news. DAN [15] considers both news title and profile information, learning the title level and profile level representation of news, and concatenation them as the final news feature representation. CAGE [16] builds an auxiliary knowledge graph to enrich the semantic meaning of entities involved in articles and further refines the article embeddings by graph convolutional networks. FUM [17] is a fine-grained and fast user modeling framework, the core idea of which is to connect the clicked news to a long document, and transform user modeling into a document modeling task with both intra-news and inter-news word-level interactions. UNBERT proposes a BERT-based user–news matching model, which explores the use of the successful BERT pre-training technique in NLP for news recommendation. DeepVT [18] focuses on the view level information of user modeling and proposes a Deep View Time Interaction Network for news recommendation, which effectively solves the problem of only focusing on a single view interaction or temporal information in user profiling modeling. MLCI [19] designs a candidate interaction module to capture competition information between multiple candidate news and extract unique features of each candidate news. GLSM [20] proposes a personalized news recommendation method based on groups, which has long-term and short-term matching mechanisms between users and PLM-based candidate news to efficiently and effectively learn fine-grained matching. DIGAT [21] models multi-level relationships between click news and enhances representation through dual-graph interaction. Although these methods have achieved performance improvements, they often lack comprehensiveness and the importance of candidate news for modeling user representations. Thus, we propose a graph neural news recommendation method based on multi-view representation learning (GRMR), which encodes higher-order information by spreading and embedding on the graph.

## 3 Problem formulation

The news recommendation problem in our paper is illustrated as follows. Given a target user $u$ and a candidate news $h_c$, our goal is to calculate the probability $y$ of clicking to predict the user's preference for candidate news. User $u$ historical clicks on the news sequence are marked as $s_u = [h_1, \ldots, h_i, h_n]$, $n$ indicates the number of clicks on the news. This paper considers the title, body and entity. Each news title contains a sequence of words $T = [w_1, w_2, w_3 ....., w_T]$, where $w_i$ denotes the i-th word in news title. Title contains the entity sequence $E = [e_1, e_2, \ldots, e_E]$, where $e_i$ denotes the i-th entity. The body is composed of a text sequence $B = [b_1, b_2, \ldots, b_B]$, where $b_i$ denotes the i-th word in news body. $T$, $B$ and $E$, respectively, denote the number of words in three sequences. The overview of our proposed GRMR approach is shown in Fig. 1. The inputs of GRMR consist of three parts: the user historical clicked news $S_u = [h_1, h_2, \ldots, h_n]$, the candidate news $h_c$ and the user–news interaction matrix $\mathbf{M}$. For each news title, we first convert its every word to a vector via an embedding matrix and convert body to a vector. Then, we can obtain the representation of the title T. For each entity, we also convert it to a vector via another embedding matrix.

## 4 Proposed method

In this section, we introduce our GRMR approach in detail. It contains four major modules, i.e., news encoder, interest encoder, aggregation and predictor.

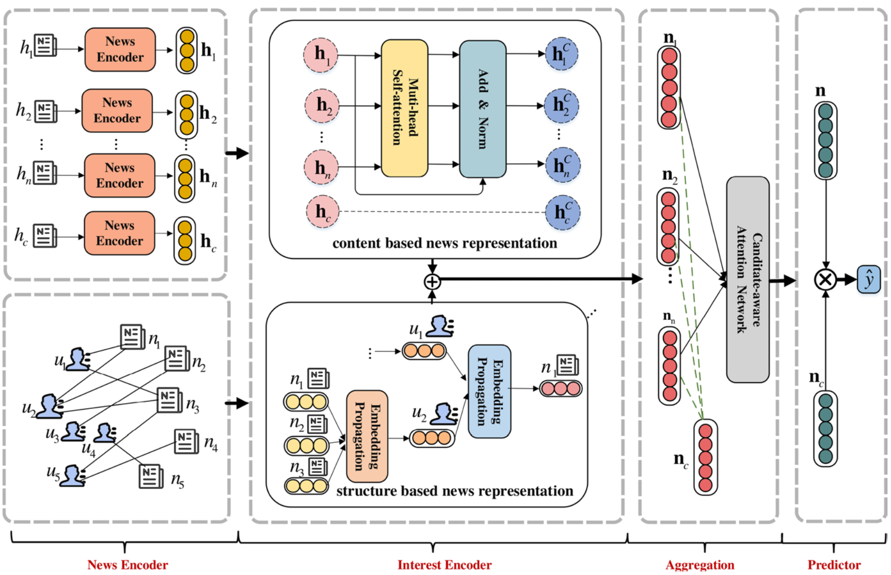


**Fig. 1** Illustration of the GRMR framework

### 4.1 News encoder

This article considers to learn news representation from various news attributes (i.e., title, body and entity). Since news titles directly determine the user's clicking behavior, we model them and learn the semantic representation of news. Each title can be represented as word embedding matrix $\mathbf{T} = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3....., \mathbf{w}_T] \in R^{T\times d}$, $\mathbf{w}_i$ denotes the i-th word embedding in news title. Then use a conventional neural network (CNN), which extracts more effective news features, incorporate the contextual features of the news word sequence. Specifically, the matrix $\mathbf{T}$ would be applied to a filter $\mathbf{H} \in R^{T\times l}$, where $l(l \ll T)$ is the window size of the filter. Thus produce a new feature map $\mathbf{m}$ as follows:

$$\mathbf{m} = f(\mathbf{T} \odot \mathbf{H} + \mathbf{b}) \tag{1}$$

where $f$ is a nonlinear function, $\odot$ is the convolution operator and $\mathbf{b} \in R^{T\times l}$ is a bias. Then we use a max-over-time pooling operation on feature map $\mathbf{m}$ to identify the most significant feature $\mathbf{f}$:

$$\mathbf{f} = max(\mathbf{m}) \tag{2}$$

Meanwhile we utilize multiple filters to obtain multiple features, and concatenate them together to obtain finally representation $\mathbf{t}$ of the news title.

Next, we briefly introduce the learning process of entity and body representation. Representation of the body is randomly initialized, and each entity is associated with its embeddings pre-trained based on the knowledge graph. Motivated by Geet al. [22], we utilize transformer and an attention network to model entities representation $\mathbf{e}$, body representation $\mathbf{b}$. Finally, we concatenate three news attributes embeddings, and we send the concatenated vector representation to a fully connected network (FNN). In this way, input sequence of click news $S_u = [h_1, h_2....., h_n]$ and candidate news $h_c$ is converted into a vector sequence $\mathbf{S}_u = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n]$ and $\mathbf{h}_c$.

### 4.2 Interest encoder

In this section, we model the representation of news from different perspectives. They are content-based news representation and structure-based news representation.

#### 4.2.1 Content-based news representation

To enhance the presentation of click news, consider the relevance between click news. There are often some potential associations between news clicked by the same user. To some extent, these associations reflect the interests of users. Capturing and modeling these underlying relationships can better enhance the presentation of click news. Inspired by [23], we use a multi-head self-attention mechanism to achieve this goal. The multi-head self-attention layer aims to capture the contextual representation of each click on the news. Given three input matrices $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$, where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ represent the query, key and value matrix correspondingly, which are projected from $\mathbf{S}_u$ with different learned projection matrices as in Eq. (4). where $d_k$

is the dimension. It applies Scaled Dot-Product Attention as the attention function to learn the combination weights of the output:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \left( \frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}} \right)\mathbf{V} \tag{3}$$

Let us take the representation of the i-th click news learned from the k-th attention head as an example to apply multi-head self-attention to model the correlation between the i-th click news and other click news. Multi-head self-attention applies h attention functions in parallel to produce the output representations:

$$head_i = Attention(\mathbf{S}_u\mathbf{W}_i^Q, \mathbf{S}_u\mathbf{W}_i^K, \mathbf{S}_u\mathbf{W}_i^V) \tag{4}$$

$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [head_1;...;head_m]\mathbf{W}^O \tag{5}$$

$\mathbf{W}_i^Q$,$\mathbf{W}_i^K$,$\mathbf{W}_i^V$,$\mathbf{W}^O$ are the projection parameters to learn, where $i$ represents $i$- $th$ head, $m$ indicates the number of heads. [;] denotes the column-wise concatenation. To fuse the original representation with the resulting context representation, we use the residual join and layer normalization function LayerNorm. The fusion process is as follows:

$$\widetilde{\mathbf{H}} = LayerNorm(\mathbf{S}_u + MultiHead(\mathbf{S}_u, \mathbf{S}_u, \mathbf{S}_u)) \tag{6}$$

where $\widetilde{\mathbf{H}} = [\mathbf{h}_1^C, \mathbf{h}_2^C, \dots, \mathbf{h}_n^C]$. Thus, we get representation $\widetilde{\mathbf{H}}$ of the input news $[h_1, ..., h_2, h_n]$ for each click of the user. We get representation $\mathbf{h}_c^C$ of the candidate news $h_c$.

#### 4.2.2 Structure-based news representation

The collaborative signals contained in users–news bipartite graph play a crucial role in optimizing the semantic representation of news. To enhance news representation learning, we propose to solve the mining of collaborative signals and the modeling of embedding based on high-order connectivity in user–item bipartite graph. We model the higher-order connection structure of the news. First, we construct a heterogeneous user–news bipartite graph by taking advantage of the users–news interaction matrix $\mathbf{M}$. In our bipartite graph, users and news are both viewed as nodes, user click behaviors are viewed as edges. Expanding clicked news $[h_1, h_2, \dots, h_n]$ and candidate news $h_c$ from users–news bipartite graph, their corresponding high-order structure graph can be obtained. Then we apply graph neural networks for learning embeddings of users and news, which encodes the high-order information between users and news through propagating embeddings over the graph.

For bipartite graphs, our model makes full use of the high-order structural information between users and news items, and then applies GNN propagation embedding, GNN can iteratively spread information from interactive users and update the news vector, which can enhance news representation. Specifically, GNN iteratively aggregates two-hop neighborhood information to encode news nodes. Motivated by [24], taking click news $n_i$ as an example, the two-order structure aggregation process after expansion is as follows:

$$\mathbf{g}_{n_i}^{k} = \sigma(\mathbf{W}_{n_i}^{k} \cdot CONCAT(\mathbf{g}_{n_i}^{k-1}, \mathbf{g}_{N(n_i)}^{k-1}), \mathbf{g}_{n_i}^{0} = \mathbf{e}_{n_i} \tag{7}$$

$$\mathbf{g}_{N(n_i)}^{k-1} = \sigma(MEAN(\left\{ \mathbf{g}_{u_i}^{k-1} \cdot \mathbf{Q}_{n_i}^{k}, u_i \in N(n_i) \right\})) \tag{8}$$

$$\mathbf{g}_{u_i}^{k} = \sigma(\mathbf{W}_{u_i}^{k} \cdot CONCAT(\mathbf{g}_{u_i}^{k-1}, \mathbf{g}_{N(u_i)}^{k-1}), \mathbf{g}_{u_i}^{0} = \mathbf{e}_{u_i} \tag{9}$$

$$\mathbf{g}_{N(u_i)}^{k-1} = \sigma(MEAN(\left\{ \mathbf{g}_{n_i}^{k-1} \cdot \mathbf{Q}_{u_i}^{k}, n_i \in N(u_i) \right\})) \tag{10}$$

where $\mathbf{e}_{n_i}$ and $\mathbf{e}_{u_i}$ are separately the initial news embeddings and user embeddings, $CONCAT$ represents concatenation, $\sigma(\cdot)$ is the tanh activation function, $\mathbf{W}_{u_i}^{k}$, $\mathbf{W}_{n_i}^{k}$ is the layer-k transformation weight matrix shared across all user nodes and news nodes. $\mathbf{g}_{N(n_i)}^{k-1}$, $\mathbf{g}_{N(u_i)}^{k-1}$ is the learned neighborhood embedding. $MEAN$ donates mean aggregator. $N(n_i)$ represents the set of neighbor users for the i-th news. $N(u_i)$ represents the set of neighbor news for the i-th users. $\mathbf{Q}_{n_i}^{k}$, $\mathbf{Q}_{u_i}^{k}$ is the layer-k aggregator weight matrix, which is shared across all user nodes and news nodes at layer k. After the above calculation, you can get a two-hop aggregate information representation of click news $\mathbf{n}_i^G$ ($\mathbf{g}_{n_i}^k$), and the same reason can get the two-hop aggregate information representation of candidate news $\mathbf{n}_c^G$ ($\mathbf{g}_c^k$). Finally, we concatenate the click news and candidate news representation correspond to the two-hop aggregate representation of click news and candidate news, which is expressed as follows:

$$\mathbf{n}_i = (\mathbf{h}_i^C;\mathbf{n}_i^G) \tag{11}$$

$$\mathbf{n}_c = (\mathbf{h}_c^C;\mathbf{n}_c^G) \tag{12}$$

Thus the new click news representation sequence $[\mathbf{n}_1, \dots, \mathbf{n}_i, \dots, \mathbf{n}_n]$ and candidate news $\mathbf{n}_c$ are obtained, where $\mathbf{h}_c^C = \mathbf{h}_c$.

### 4.3 Aggregation

In this module, we will form the user interest representations by aggregating the news representations clicked by the user. User interest in news topics may vary, and news clicked by the user is considered to have a different impact on candidate news. In order to characterize the different interests of users, we use candidate-aware attention to simulate the different effects of click news on candidate news. It automatically matches each clicked news to candidate news and aggregates the user's current interest with different weights. Specifically, according to user click news embedding representation sequence$[\mathbf{n}_1, \ldots, \mathbf{n}_i, \ldots, \mathbf{n}_n]$ and candidate news embedding representation $\mathbf{n}_c$, user's current interest feature representations $\mathbf{n}$ is calculated as follows:

$$\mathbf{v}_i = \tanh(\mathbf{P}_w \cdot \mathbf{n}_i + \mathbf{b}_w) \tag{13}$$

$$\mathbf{v}_c = \tanh(\mathbf{P}_c \cdot \mathbf{n}_c + \mathbf{b}_c) \tag{14}$$

$$\alpha_{i,c} = \frac{exp(\mathbf{a}^\top(\mathbf{v}_i + \mathbf{v}_c))}{\sum_i exp(\mathbf{a}^\top(\mathbf{v}_i + \mathbf{v}_c))} \tag{15}$$

$$\mathbf{n} = \sum_i \alpha_{i,c} \cdot \mathbf{n}_i \tag{16}$$

where $\mathbf{P}_w, \mathbf{b}_w, \mathbf{P}_c, \mathbf{b}_c$ represent the parameters in the attention mechanism, respectively; $\mathbf{a}$ represents a query vector; $\alpha_{i,c}$ indicates the weight of the impact of a user's click on a news candidate news.

**Algorithm 1** News Recommendation Process

**Require:** the user historical clicked news $S_u = [h_1, h_2, ..., h_n]$, the candidate news $h_c$, the user-news interaction graph G=(V,E), $|V|$ is the node numbers.
**Ensure:** the click probability score $\hat{y}$

Initialize all nodes embedding and model parameters in graph G;
**for** i=1 to n **do**
  $\mathbf{h}_i \leftarrow NewsEncoder(h_i)$;
**end for**
$\mathbf{h}_c \leftarrow NewsEncoder(h_c)$;
$\mathbf{S}_u = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n]$
$\widetilde{\mathbf{H}} \leftarrow MutiheadSelfattention(\mathbf{S}_u)$
$\widetilde{\mathbf{H}} = [\mathbf{h}_1^C, \mathbf{h}_2^C, ..., \mathbf{h}_n^C]$
**for** i=1 to $|V|$ **do**
  $\mathbf{g}_{n_i}^k = \sigma(\mathbf{W}_{n_i}^k \cdot CONCAT(\mathbf{g}_{n_i}^{k-1}, \mathbf{g}_{N(n_i)}^{k-1}), \mathbf{g}_{n_i}^0 = \mathbf{e}_{n_i}$;
  $\mathbf{g}_{u_i}^k = \sigma(\mathbf{W}_{u_i}^k \cdot CONCAT(\mathbf{g}_{u_i}^{k-1}, \mathbf{g}_{N(u_i)}^{k-1}), \mathbf{g}_{u_i}^0 = \mathbf{e}_{u_i}$
**end for**
**for** i=1 to n **do**
  $\mathbf{n}_i = \mathbf{h}_i^C || \mathbf{n}_i^G$
**end for**
$\mathbf{n}_c \leftarrow \mathbf{h}_c || \mathbf{n}_c^G$
$\mathbf{n} \leftarrow CanditateawareAttentionNetwork(\mathbf{n}_i; \mathbf{n}_c)$
$\hat{y} \leftarrow \mathbf{n} \cdot \mathbf{n}_c$
**return** $\hat{y}$

### 4.4 Predictor and training

For each candidate news $\mathbf{n}_c$, we calculate a matching score with the user representation $\mathbf{n}$ via inner product:

$$\hat{y} = \mathbf{n} \cdot \mathbf{n}_c \tag{17}$$

Then we train models based on the BPR [25] loss:

$$\mathcal{L} = -\frac{1}{|D|} \sum_{i=1}^{|D|} log\sigma(\hat{y}_i^p - \hat{y}_i^n) \tag{18}$$

where D is the training dataset, $\sigma$ is sigmoid function, $\hat{y}_i^p$ and $\hat{y}_i^n$ are the matching score of the i-th positive and negative sample.

The news recommendation algorithm infers representation learning process of news and users. We use the user historical clicked news $S_u = [h_1, h_2, ..., h_n]$, the candidate news $h_c$, the user–news interaction graph G=(V,E) as input. We use NewsEncoder to learn click news and candidate news representation. Then we use MultiheadSelfattention to enhance these representation. For user–news interaction graph G, we learn the representation of news node in the graph through GNN. Then concatenates the representations obtained from the two parts. Finally, we use CanditateawareAttentionNetwork to learn user representation. Based on this, we can get the probability that a user clicks on a candidate news. The detail is illustrated in Algorithm 1.

## 5 Results

### 5.1 Datasets and experimental settings

To assess the performance of the proposed method, we conduct experiments on a real-world online news dataset Adressa [26], which is a click log dataset with approximately 20 million page visits from a Norwegian news portal as well as a sub-sample with 2.7 million clicks. Adressa is published with the collaboration of Norwegian University of Science and Technology (NTNU). The second one is the public MIND datasets [27]. It is constructed by user behavior data collected from Microsoft News from October 12 to November 22, 2019 (six weeks). It contains MIND and MIND-small versions, of which MIND contains 161,031 news, 1 000,000 users and 24,155,470 behavioral logs; each news article contains a headline, summary, body, category and entity; in each behavior log, click on the item, not click and the day of the behavior. History before the history click news; lightweight MIND-small package contains 93,698 news, 50,000 users and 230117 actions log. Table 1 summarizes the statistics of the above two datasets. In our experiments, word embeddings were 300-dimensional, we used Glove [28] to initialize the word embeddings. The attention networks had 16 heads, and each head was 16-dimensional. The attention query was 200-dimensional. The intensity of dropout

**Table 1** Statistics of datasets

| Items | MIND | Adressa |
|---|---|---|
| Language | English | Norwegian |
| #Users | 1,000,000 | 3,083,438 |
| #News | 161,013 | 48,486 |
| #Clicks | 24,155,470 | 27,223,576 |
| #News Information | Title, abstract, body, category | Title, body, category |

was 20%. The batch size was set to 32. These hyper-parameters were tuned on the validation set. Each experiment was repeated 10 times to mitigate occasional.

### 5.2 Evaluation metrics

Following previous works, we use average AUC (Area Under Curve), Mean Reciprocal Rank (MRR), nDCG@5(Normalized Discounted Cumulative Gain) and nDCG@10 to evaluate model performance.

AUC refers to the area enclosed by the ROC curve and the X coordinate axis. The calculation formula for AUC is derived based on the definition of AUC:

$$AUC = \frac{\sum(p_i, p_j)_{p_i > p_j}}{P * N} \tag{19}$$

Where $P$ is the number of positive samples, $N$ is the number of negative samples, $P_i$ is the prediction score of positive samples and $n_j$ is the prediction score of negative samples.

MRR evaluates the performance of the search system by correctly ranking the result values in the search results. The calculation formula is as follows:

$$MRR = \frac{1}{Q} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{20}$$

where $|Q|$ is the number of users. $rank_i$ is the ranking position of the first item in the ground-truth result in the recommended list for the i-th user.

$$NDCG = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{log_2(p_i + 1)} \tag{21}$$

Where N represents the total number of users, $p_i$ represents the position of the i-th user's actual access value in the recommendation list. If the value does not exist in the recommendation list, then $p_i \rightarrow \infty$.

### 5.3 Performance evaluation

We use the following state-of-the-art methods as baselines in our experiments.

DKN [29] is a content-based deep recommendation framework for click-through rate prediction. The key component of DKN is a multi-channel and word-entity-aligned knowledge-aware convolutional neural network (KCNN) that fuses semantic-level and knowledge-level representations of news.

NAML [23] proposes a neural news recommendation approach which can learn informative representations of users and news by exploiting different kinds of news information. The core of approach is a news encoder and a user encoder.

CAUM [30] incorporates candidate news into user modeling to better match candidate news and user interests. It uses candidate news as a clue to model candidate-aware of global user interest. And the candidate news is integrated into the local behavior context modeling.

FUM [17] models user interests from fine-grained behavioral interactions for news recommendation. The core idea of FUM is to connect clicked news into a long document, and to transform user modeling into a document modeling task with word-level interactions within and between news.

GLSM [21] proposes a group-based personalized news recommendation method, which establishes a long-term and short-term matching mechanism between users and candidate news on the basis of PLM, so as to learn fine-grained matching efficiently and effectively.

GERL [22] enhances the representation learning of users and news by modeling their relatedness in a graph setting.

We repeat experiments of different methods 5 times on MIND and Adressa datasets and show average results and standard deviations in Tables 2 and 3. From the table, we have the following observations. DKN and NAML are two content-based methods. Although these content-based methods have achieved good performance, due to the fact that a lot of new news is generated every moment, user–news interactions are very sparse, and there may be a cold start issue, which seriously affects the performance of these models. When modeling user representations, CAUM uses candidate news as guidance, which can effectively match candidate news with user interests. But its disadvantage is that it may suffer from data sparsity issues. FUM considers the structural information between users and click news, but does not use

**Table 2** News recommendation performance of different methods on MIND

| Model | AUC (%) | MRR (%) | nDCG@5 (%) | nDCG@10 (%) |
|---|---|---|---|---|
| DKN | 64.08±0.12 | 29.06±0.16 | 31.82±0.11 | 38.52±0.14 |
| NAML | 64.28±0.27 | 29.77±0.21 | 32.10±0.28 | 39.75±0.24 |
| CAUM | 65.08±0.12 | 30.06±0.16 | 33.82±0.11 | 38.97±0.14 |
| FUM | 66.28±0.27 | 31.77±0.21 | 33.10±0.28 | 39.96±0.24 |
| GLSM | 66.82±0.21 | 32.95±0.11 | 34.12±0.09 | 40.98±0.12 |
| GERL | 67.08±0.11 | 33.05±0.13 | 34.86±0.14 | 41.52±0.04 |
| GRMR | 67.21±0.04 | 33.25±0.11 | 35.34±0.13 | 41.99±0.20 |

**Table 3** News recommendation performance of different methods on Adressa

| Model | AUC (%) | MRR (%) | nDCG@5 (%) | nDCG@10 (%) |
|---|---|---|---|---|
| DKN | 62.91±0.26 | 28.08±0.20 | 32.20±0.24 | 37.75±0.22 |
| NAML | 64.24±0.38 | 28.81±0.21 | 33.06±0.28 | 38.52±0.29 |
| CAUM | 64.99±0.21 | 29.88±0.16 | 33.98±0.22 | 38.02±0.10 |
| FUM | 65.87±0.10 | 30.02±0.14 | 34.86±0.11 | 38.88±0.05 |
| GLSM | 66.01±0.02 | 30.03±0.10 | 35.12±0.04 | 38.99±0.01 |
| GERL | 66.91±0.06 | 30.98±0.120 | 35.99±0.04 | 39.55±0.02 |
| GRMR | 67.12±0.01 | 31.18±0.11 | 36.64±0.12 | 39.99±0.09 |

candidate news as guidance and does not utilize user interest mining. GLSM makes news recommendations from other perspectives, which is slightly inferior to our method. The performance of GERL has been improved due to the consideration of higher-order structural relationships. GRMR considers the content information of news from various perspectives while considering high-order relationships, and uses candidate news as guidance to model user interests. Its performance is superior to other baseline methods.

## 5.4 Ablation study

In this section, to investigate the effectiveness of different components in our GRMR framework, we conduct an ablation study. First, we assess the effectiveness of GNN (i.e., graph neural networks) in modeling news high-order structure information. We used three evaluation metrics NDCG, nDCG@5 and nDCG@10 to illustrate the experimental results. The experimental results are shown in Figs. 2 and 3, from which we have several observations. Removing high-order structure information (i.e., GNN) can seriously harm the performance of GRMR. This is because in the

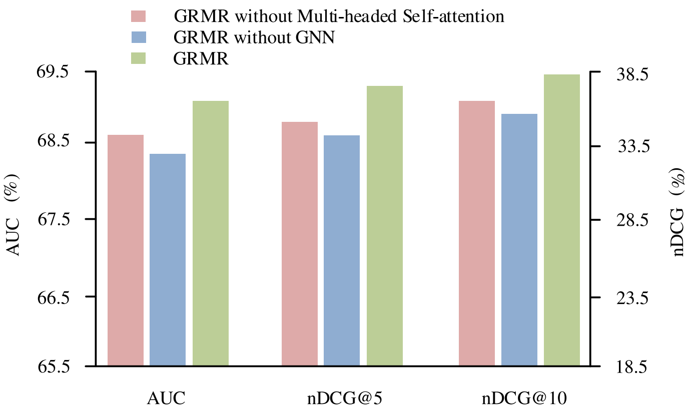


**Fig. 2** Ablation study of GRMR on the MIND dataset

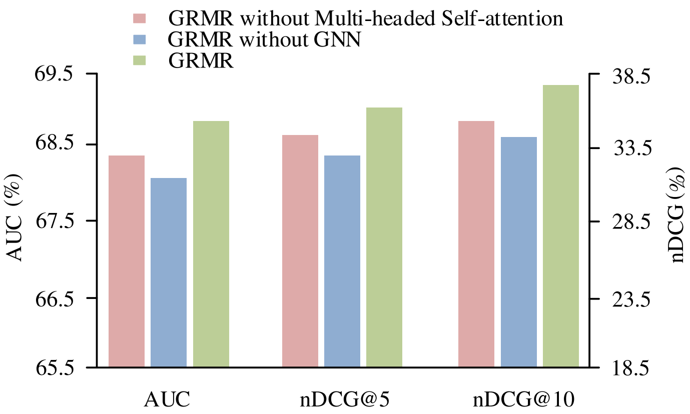


**Fig. 3** Ablation study of GRMR on the Adressa dataset

user–news click network, the user information that clicks on different news is also very important for modeling news representation. Click relationships between user and news often contain a wealth of information, which is essential for the understanding of the news. Removing high-order structure information causes the news representation to lose a lot of important information and make it impossible to accurately model the news representation. Secondly, we evaluated the importance of the multi-head self-attention in modeling the potential relationship between click news. When removing the multi-head self-attention, the performance of the model decreases, indicating that multi-head self-attention can capture the potential relationship of clicked news and enhance news representation.

## 5.5 Influence of hyper-parameters

### 5.5.1 The number of GNN layers

We evaluate an important hyper-parameter, the number of GNN layers. In order to verify the influence of variables on the experimental results, this section gives the influence of the number of GNN layers and gives the change curve of the influence of hyper-parameters on the model performance. The influence of different GNN layers on the experimental results is shown in Fig. 4. Analyzing the experimental data, we have the following observations. First, the performance of the model first improves as the number of layers K of convolution increases. This is because there is an important potential relationship between the user's multiple click news, and the connection between these click news contains important knowledge that reflects the user's interests. When K is too small, it can only be simply convoluted to the user's information, and the correlation between the user and other clicked news and candidate news cannot be fully explored, which is not conducive to the accuracy of the recommendation. Second, when K is too large, the performance of GRMR begins to degrade. This is because when K becomes too large, too many multi-hop neighbors are considered when modeling the

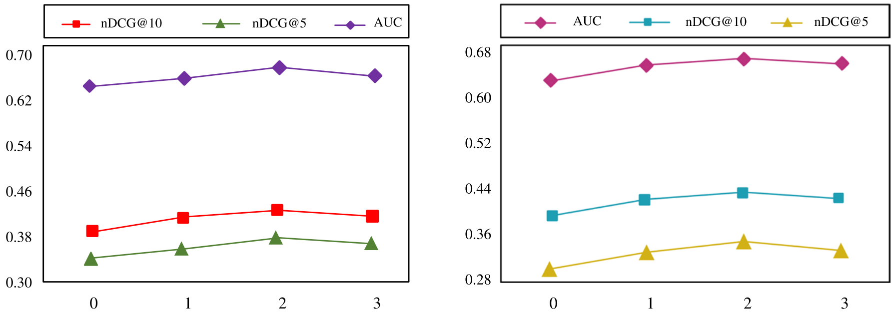


**Fig. 4** Effect of GNN layers on model performance on MIND and Adressa datasets

higher-order connection structure of the user's news click relationship. This will accumulate a lot of useless information, which will harm the performance of the model. Thus, a moderate value of K, i.e., 2, is suitable for GRMR.

### 5.5.2 Dimension of word embedding *d* and entity embedding *k*

We select the test sets 25, 55, 105, 205 and use all combinations of d and k to study the effect of the dimension of the word embedding d and the dimension of the entity embedding $k$ on performance. The results are shown in Fig. 5. We can get the following observations. Given the entity dimension $k$, when the value of the word embedding dimension d increases, the value of AUC also increases, and the performance of the model improves. This is because larger dimensions can enhance the expressiveness of vector spaces. However, when the value of d increases by more than 105, the value of AUC will decrease, because too large dimensions will introduce noise, and there may be overfitting cases, which affect the expression ability of vectors. The case is similar for k when d is given.

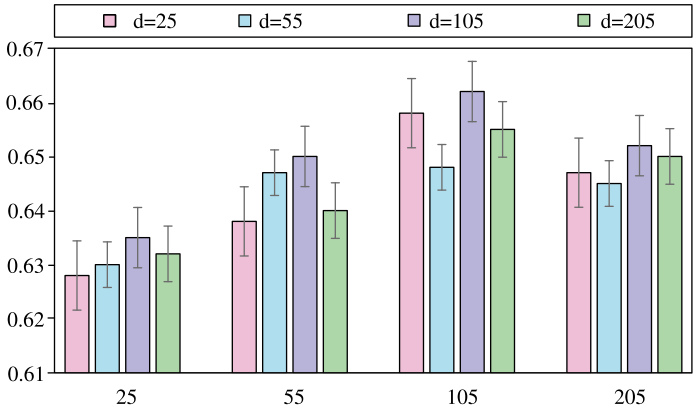


**Fig. 5** AUC score w.r.t dimension of entity embedding $k$ and dimension of word embedding $d$

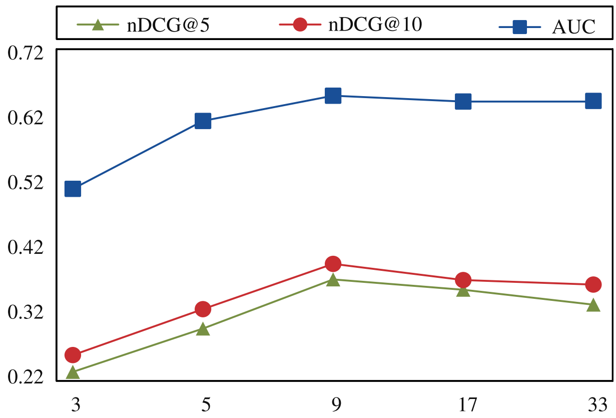


**Fig. 6** Performance of GRMR under different number of attention heads

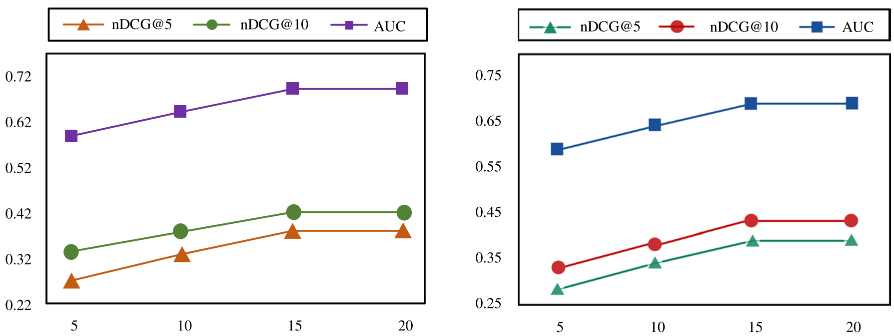


**Fig. 7** Performance of GRMR under different number of neighbor nodes on MIND and Adressa datasets

### 5.5.3 The number of attention head

We evaluate the impact of the number of attention heads in multi-head self-attention on model performance. The number of self-attention heads is important for learning contextual content. The experimental results are shown in Fig. 6. From it, we can conclude that as the number of heads increases, the values on the three evaluation metric first increase and then decrease. This is because more headers can capture more context information, but when the number exceeds a certain value, too many parameters will be introduced, which can harm the performance of the model. Based on this, we set the number to 9.

### 5.5.4 The number of neighbor nodes

The number of neighbor nodes determines the amount of information aggregated by the target node and has a significant impact on the quality of the vector representation of the target node. The experimental results are shown in Fig. 7, from which we can draw the conclusion that, in the initial period, as the number of neighbor nodes increases, the performance of the model improves on three evaluation metrics. However, when the number of neighbor nodes increases to a certain value, the performance improvement of the model becomes slow. This is because an appropriate number of neighbor nodes does enhance the vector representation of the target node, but when the number exceeds a certain value, noise data will be introduced, impairing the performance of the model.

## 5.6 Case study

To intuitively demonstrate the efficacy of our model, we conduct a case study to show the effectiveness of GRMR by comparing it with GERL and FIM. We chose them because they achieve the better performance (Tables 2 and 3) in the baseline method. The experimental results are shown in Fig. 8. We randomly pick a user and pull out the first few news items he clicked on. At the same time, we show the top news recommended by the three methods listed, and the top-ranked news is the candidate news that users click on, from which we have several observations. First, GRMR gets higher rankings for click-on news candidates than GERL and FIM. This is because GRMR not only considers the text content information of the news but also considers the high-level structure information, and integrates

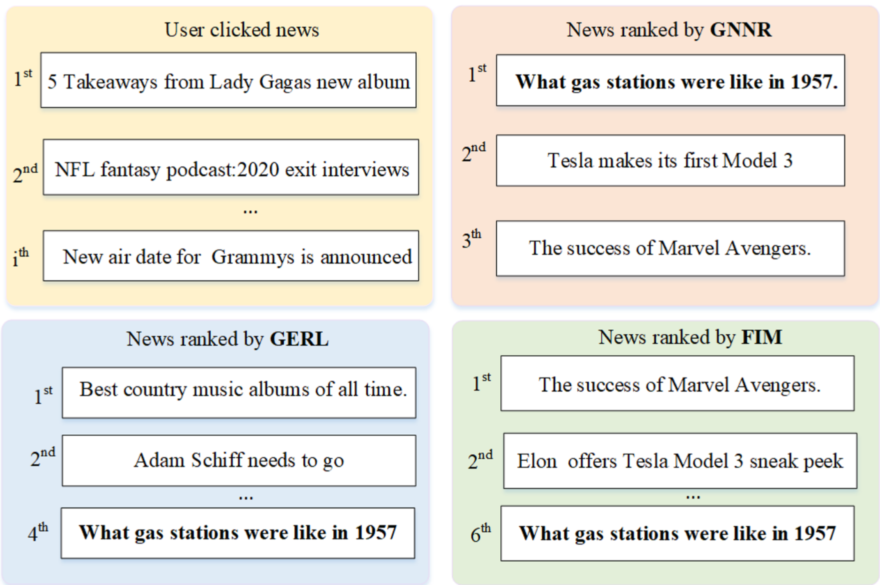


**Fig. 8** News recommended to a randomly selected user by different methods. The news in blue is the news actually clicked by this user in this impression. The historical clicked news of this user are also shown in this figure (color figure online)

the potential information in the user's news click map into the news representation, improving the quality of the news representation.

## 6 Conclusion

We propose a graph neural news recommendation with candidate-aware attention network GRMR. Our model constructs a heterogeneous user–news graph to model user–item interactions. Then it applies graph conventional networks to learn user and news embeddings with high-order information encoded by propagating embeddings over the graph. This allows node information and topology to be integrated naturally, and dependencies between nodes can be effectively modeled. In addition, to address users' diverse interests, we also design an attention module to dynamically aggregate a user's history with respect to current candidate news. Extensive experiments on benchmark datasets show that our approach can effectively improve performance in news recommendation.

**Author contributions** X.L and R.L were involved in conceptualization, methodology, formal analysis, software, investigation, validation, resources, writing—original draft, review and editing and visualization. Q.P and J.Y were responsible for resources, writng—review and editing and supervision.

**Funding** No funding was received to assist with the preparation of this manuscript.

**Availability of data and materials** The datasets analyzed during the current study were all derived from the following public domain resources.[https://msnews.github.io/;http://reclab.idi.ntnu.no/dataset/].

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Consent to participate** The authors declare that they agree to participate.

**Consent for publication** The authors declare that they agree to publish.

## References

1. Li L, Chu W, Langford J, Schapire RE (2010) A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp 661–670
2. Liu J, Dolan P, Pedersen ER (2010) Personalized news recommendation based on click behavior. In: Proceedings of the 15th International Conference on Intelligent User Interfaces, pp 31–40
3. Phelan O, McCarthy K, Smyth B (2009) Using twitter to recommend real-time topical news. In: Proceedings of the Third ACM Conference on Recommender Systems, pp 385–388
4. Kazai G, Yusof I, Clarke D (2016) Personalised news and blog recommendations based on user location, facebook and twitter user profiling. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 1129–1132
5. Karkali M, Pontikis D, Vazirgiannis M (2013) Match the news: a firefox extension for real-time news recommendation. In: Proceedings of the 36th International Acm Sigir Conference on Research and Development in Information Retrieval, pp 1117–1118

6. Wang C, Blei DM (2011) Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 448–456
7. IJntema W, Goossen F, Frasincar F, Hogenboom F (2010) Ontology-based news recommendation. In: Proceedings of the 2010 EDBT/ICDT Workshops, pp 1–6
8. De Francisci Morales G, Gionis A, Lucchese C (2012) From chatter to headlines: harnessing the real-time web for personalized news recommendation. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, pp 153–162
9. Kelly D, Teevan J (2003) Implicit feedback for inferring user preference: a bibliography. ACM Sigir Forum, vol 37. ACM New York, NY, USA, pp 18–28
10. Peng Y-x, Zhu W-w, Zhao Y, Xu C-s, Huang Q-m, Lu H-q, Zheng Q-h, Huang T-j, Gao W (2017) Cross-media analysis and reasoning: advances and directions. Front Inf Technol Electron Eng 18(1):44–57
11. An M, Wu F, Wu C, Zhang K, Liu Z, Xie X (2019) Neural news recommendation with long-and short-term user representations. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp 336–345
12. Wang H, Wu F, Liu Z, Xie X (2020) Fine-grained interest matching for neural news recommendation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp 836–845
13. Zhang H, Chen X, Ma S (2019) Dynamic news recommendation with hierarchical attention network. In: 2019 IEEE International Conference on Data Mining (ICDM), pp 1456–1461 IEEE
14. Wu C, Wu F, An M, Huang J, Huang Y, Xie X (2019) Npa: neural news recommendation with personalized attention. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 2576–2584
15. Zhu Q, Zhou X, Song Z, Tan J, Guo L (2019) Dan: deep attention neural network for news recommendation. Proc AAAI Conf Artif Intell 33:5973–5980
16. Sheu H-S, Li S (2020) Context-aware graph embedding for session-based news recommendation. In: Proceedings of the 14th ACM Conference on Recommender Systems, pp 657–662
17. Qi T, Wu F, Wu C, Huang Y (2022) Fum: fine-grained and fast user modeling for news recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 1974–1978
18. Zhang X, Yang Q, Xu D (2022) Deepvt: deep view-temporal interaction network for news recommendation. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management
19. Sun Y, Kong Q, Mao W, Tang S (2022) Multi-level news recommendation via modeling candidate interactions. In: 2022 7th International Conference on Big Data Analytics (ICBDA), pp 290–295. IEEE
20. Xu H, Peng Q, Liu H, Sun Y, Wang W (2023) Group-based personalized news recommendation with long- and short-term fine-grained matching. ACM Trans Inf Syst
21. Xu H, Peng Q, Liu H, Sun Y, Wang W (2023) Group-based personalized news recommendation with long-and short-term fine-grained matching. ACM Trans Inf Syst
22. Ge S, Wu C, Wu F, Qi T, Huang Y (2020) Graph enhanced representation learning for news recommendation. Proc Web Conf 2020:2863–2869
23. Wu C, Wu F, An M, Huang J, Huang Y, Xie X (2019) Neural news recommendation with attentive multi-view learning. arXiv preprint arXiv:1907.05576
24. Sun J, Zhang Y, Ma C, Coates M, Guo H, Tang R, He X (2019) Multi-graph convolution collaborative filtering. In: 2019 IEEE International Conference on Data Mining (ICDM), pp 1306–1311 . IEEE
25. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) Bpr: Bayesian personalized ranking from implicit feedback. arXiv:1205.2618
26. Gulla J.A, Zhang L, Liu P, Özgöbek Ö, Su X (2017) The adressa dataset for news recommendation. In: Proceedings of the International Conference on Web Intelligence, pp 1042–1048
27. Wu F, Qiao Y, Chen J-H, Wu C, Qi T, Lian J, Liu D, Xie X, Gao J, Wu W, et al (2020) Mind: a large-scale dataset for news recommendation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp 3597–3606
28. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp 1532–1543

29. Wang H, Zhang F, Xie X, Guo M (2018) Dkn: deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide Web Conference, pp 1835–1844
30. Qi T, Wu F, Wu C, Huang Y (2022) News recommendation with candidate-aware user modeling. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 1917–1921