DIGAT: Modeling News Recommendation with Dual-Graph Interaction

Anonymous ACL submission

Abstract

News recommendation is essential for online news applications. Existing news recommendation approaches typically adopt a two-tower encoder framework, facing two potential limitations. First, in news encoder tower, single candidate news encoding suffers from an *insufficient semantic information* problem. Second, existing graph learning models for news recommendation are promising but lack effective news-user interaction modeling, which causes the graph modeling suboptimal. To overcome these limitations, we propose dual-interactive graph attention networks (DIGAT) consisting of news- and user-graph channels. In the news-graph channel, we use a semantic-augmented graph to enrich the semantics of the single candidate news by incorporating the semantic information of relevant news. In the user-graph channel, we utilize a news-topic graph to precisely model user interests. Most importantly, we design a dual-graph interaction mechanism to model effective feature interaction between the news and user graphs, which facilitates accurate news-user representation matching. Experiment results on the benchmark dataset *MIND* show that DIGAT outperforms the existing news recommendation methods. Further ablation studies and analyses validate the effectiveness of semantic-augmented graph encoding and dual-graph interaction.

1 Introduction

News recommendation is an important technique to provide people with the news, which satisfies their personalized reading interests (Okura et al., 2017; Wu et al., 2020). Effective news recommendation systems require both accurate textual modeling on news content (Wang et al., 2018; Wu et al., 2019d; Wang et al., 2020) and personal-interest modeling on user behavior (Hu et al., 2020b; Qi et al., 2021c). In consequence, most neural news recommendation models (An et al., 2019; Wu et al., 2019a,b,c,d; Ge et al., 2020; Qi et al., 2021a,b,c) adopt a two-tower encoder framework to learn fine-grained news and user representations, as illustrated in Figure 1.

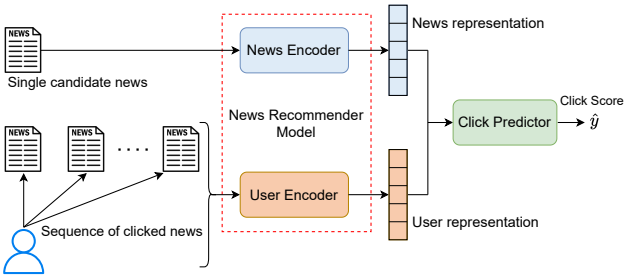


Figure 1: The common two-tower encoder framework for news recommendation.

Though it is promising, there are still two potential limitations in the two-tower encoder framework. First, in news encoder tower, single candidate news encoding suffers from an *insufficient semantic information* problem. Unlike *long-term* items in common recommendation (e.g., E-commerce product recommendation), the candidate news items in news recommendation are *short-term* and lack sufficient user-clicks. In the real-world setting, news recommendation systems usually handle the latest news, where existing user-click interactions are always not available[1]. Hence, it is intractable to use existing user-click interactions to enrich candidate news information. On the other hand, compared to abundant click history in user encoder tower, the single candidate news may not contain sufficient semantic information for accurate news-user representation matching in the click prediction stage. Prior studies (Wu et al., 2019a,c; Qi et al., 2021c) pointed out that users were usually interested in certain news topics (e.g., *Sports* topic). Empirically, the text of single candidate news does not contain enough syntactic and semantic information to accurately match user interest in a news topic.

Second, previous studies generally follow two research directions to model user history, i.e., se-

[1]From the viewpoint of experimental dataset, most candidate news in test data does not appear in training user history.

quence and graph modeling. Formulating user history as a sequence of user's clicked news is a more prevalent direction, based on which time-sequential models (Okura et al., 2017; An et al., 2019; Qi et al., 2021b) and attentive models (Zhu et al., 2019; Wu et al., 2019a,b,d; Qi et al., 2021a,c) are proposed. Graph modeling is proved effective for recommendation systems (Chen et al., 2020). Ge et al. (2020) and Hu et al. (2020b) formulate news and users jointly in a bipartite graph to model news-user interaction. However, most candidate news in test data has no existing interaction with users, which can only be treated as isolated nodes and causes the bipartite graph modeling degenerate. Recent works formulate user history as heterogeneous graphs and employ advanced graph learning methods to extract the user-graph representations (Hu et al., 2020a; Wu et al., 2021). These works focus on how to extract fine-grained representations from the user-graph side, but neglect the necessary feature interaction between candidate news and user-graphs.

In this work, we propose **D**ual-**I**nteractive **G**raph **AT**tention networks (DIGAT) to address the aforementioned limitations. DIGAT consists of the news- and user-graph channels to encode the candidate news and user history, respectively. In the news-graph channel, we encode the single candidate news with the semantic-augmented graphs (SAG) to enrich its semantic representation. In SAG, the original candidate news is regarded as the root node, while the semantic-relevant news is regarded as the extended node to augment the semantics of the candidate news. We integrate the local and global contexts of SAG as the semantic-augmented candidate news representations.

In the user-graph channel, motivated by Hu et al. (2020a) and Wu et al. (2021), we model user history with a news-topic graph to encode multi-levels of user interests. Most importantly, we design a dual-graph interaction module to learn news- and user-graph representations with effective feature interaction. Different from the individual graph attention network (Veličković et al., 2018), DIGAT updates news and user graph embeddings with the interactive attention mechanism. Particularly, in each layer of the dual-graph, the user (news) graph context is incorporated into its dual news (user) graph embedding learning iteratively.

Extensive experiments on the benchmark dataset *MIND* (Wu et al., 2020) show that DIGAT significantly outperforms the existing news recommendation methods. Further ablation studies and analyses confirm that semantic-augmented graph encoding and dual-graph interaction can substantially improve news recommendation performance.

2 Related Work

Personalized news recommendation is important to online news services (Okura et al., 2017; Yi et al., 2021). Existing news recommendation methods typically employ the two-tower encoder framework to learn news and user representations (Wang et al., 2018; Zhu et al., 2019; An et al., 2019; Wu et al., 2019a,b,d; Wang et al., 2020; Qi et al., 2021a,b,c; Wu et al., 2021). For example, An et al. (2019) used a CNN network to extract textual representation from news titles, and used a GRU network to learn short-term user interests combined with long-term user embeddings. The matching probabilities between candidate news and users are computed over the learned news and user representations. Wu et al. (2019d) utilized multi-head self-attention networks to learn informative news and user representations from news titles and user clicked history. These methods regarded the single candidate news as the input to news encoder, which may not contain sufficient semantics to represent a user-interested news topic. Different from these methods, we encode the candidate news with semantic-augmented graphs to enrich its semantic representations. More recently, graph-based methods were proposed for news recommendation (Ge et al., 2020; Hu et al., 2020a,b; Wu et al., 2021). For example, Wu et al. (2021) proposed a heterogeneous graph pooling method to learn accurate user interest representations. However, feature interaction between candidate news and users is inadequate or neglected in these methods. In contrast, our approach models effective feature interaction between news and user graphs for accurate news-user representation matching.

3 Approach

Problem Formulation. Denote the clicked-news history of a user u as $H_u = [n_1, n_2, ..., n_{|H|}]$, containing $|H|$ clicked news items. For the news n, its textual content consists of a sequence of $|T|$ words as $T_n = [w_1, w_2, ..., w_{|T|}]$. Based on H_u and T_n, the goal of news recommendation is to predict the score $\hat{s}_{n,u}$, which indicates the probability of the user u clicking the candidate news n_{can}. The recommendation result is generated by ranking the user-click scores of multiple candidate news items.

3.1 News Semantic Representation

We introduce how to extract semantic representation from news content text $T_n = [w_1, w_2, ..., w_{|T|}]$. Our news encoder first maps the news word tokens into word embeddings $E_n = [e_1, e_2, ..., e_{|T|}]$. Then, we utilize a convolutional neural network $\text{Conv}(\cdot)$ to extract the local semantic features of the news word embeddings E_n. Finally, we employ an attention network $f_{att}(\cdot)$ to aggregate the global semantic news representation as h:

$$h = f_{att}\Big(\sigma\big(\text{Conv}([e_1, e_2, ..., e_{|T|}])\big)\Big) \quad (1)$$

, where σ is ReLU activation and $h \in \mathbb{R}^d$ (d is the number of CNN feature maps). The attention function $f_{att}(\cdot)$ is implemented by a feed-forward network in our experiments. It is worth noting that the CNN news encoder can be easily replaced by any other textual encoders, e.g., Transformer[2] (Vaswani et al., 2017), or pretrained language encoders, e.g., BERT (Devlin et al., 2019).

3.2 News Graph Encoding Channel

In this section, we will explain the news semantic-augmented graph (SAG) construction and graph context learning. Our motivation is to retrieve semantic-relevant news from training corpus and construct a semantic-augmented graph to enrich the semantics of the original candidate news.

3.2.1 News Graph Construction

Semantic-relevant News Retrieval. Pretrained language models (PLM) have achieved remarkable performance (Reimers and Gurevych, 2019; Song et al., 2020) on semantic textual similarity (STS) benchmark. Motivated by Lewis et al. (2020), we utilize a PLM $\phi(\cdot)$ to retrieve semantic-relevant news from the training news corpus to augment the semantics of the single candidate news[3]. In the retrieval process, the semantic similarity score $s_{i,j}$ of news n_i and n_j (corresponding texts T_i and T_j) is computed by the similarity function $sim(\cdot, \cdot)$:

$$s_{i,j} = sim\big(n_i, n_j\big) = \text{cosine}\big(\phi(T_i), \phi(T_j)\big) \quad (2)$$

Semantic-augmented Graph. For the original candidate news n_{can}, we initialize it as the root node v_0 of the semantic-augmented news graph G_n. We build G_n by repeatedly extending semantic-relevant neighboring nodes to existing nodes of G_n. Specifically, in each graph construction step, for an existing node v_i (correspoding news N_i) of G_n, M news documents $\{N_j\}_{j=1}^M$ are retrieved from the training news corpus with the highest semantic similarity scores $\{s_{i,j}\}_{j=1}^M$. We extend the nodes $\{v_j\}_{j=1}^M$ as neighboring nodes to the node v_i by adding bidirectional edge $\{e_{i,j}\}_{j=1}^M$ between them. To heuristically discover semantic-relevant news in higher-order relation, we repeatedly extend the semantic-relevant news nodes within K hops from the root node. The scale of news graph G_n is approximated to be $\mathcal{O}(M^K)$. Detailed SAG construction and examples are provided be in Appendix A.

3.2.2 News Graph Context Extraction

Given an SAG G_n generated from the candidate news node v_0 with N semantic-relevant news nodes $\{v_i\}_{i=1}^N$, we use the semantic news encoder (described in Section 3.1) to extract their semantic representations as $h_{n,0} \in \mathbb{R}^d$ and $\{h_{n,i}\}_{i=1}^N \in \mathbb{R}^{N \times d}$.

We aim to extract the graph context $c_n \in \mathbb{R}^d$, which augments the semantics of the candidate news n_{can} by aggregating the information of G_n. We consider the original semantics of the candidate news preserved in the root node v_0 and regard the local graph context as $h_n^L = h_{n,0} \in \mathbb{R}^d$. Besides, we employ an attention module to aggregate the global graph context $h_n^G \in \mathbb{R}^d$ from the semantic-relevant news nodes to encode the overall semantic information of G_n. In the attention module, we regard the root node embedding $h_{n,0}$ as the query and the semantic-relevant news node embeddings $\{h_{n,i}\}_{i=1}^N$ as the key-value pairs.

$$e_i = \frac{(h_{n,0}\mathbf{W}_n^Q)(h_{n,i}\mathbf{W}_n^K)^T}{\sqrt{d}} \quad (3)$$

$$\alpha_i = \text{softmax}(e_i) = \frac{\exp(e_i)}{\sum_{j=1}^N \exp(e_j)} \quad (4)$$

$$h_n^G = \sum_{i=1}^N \alpha_i h_{n,i} \quad (5)$$

, where $\mathbf{W}_n^Q \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_n^K \in \mathbb{R}^{d \times d}$ are parameter matrices. We concatenate the local and global graph contexts and employ a feed-forward network to learn the news graph context c_n.

$$c_n = \sigma\big(\mathbf{W}_n[h_n^L; h_n^G] + \mathbf{b}_n\big) \quad (6)$$

, where $\mathbf{W}_n \in \mathbb{R}^{d \times 2d}$ and $\mathbf{b}_n \in \mathbb{R}^d$ are learnable parameters. Above parameters are shared among different graph layers described in Section 3.4.

[2] We empirically find that performance of the CNN encoder is slightly better than Transformer in our DIGAT framework.

[3] Specifically, we use pretrained mpnet-base-v2 (Song et al., 2020) in https://www.sbert.net/docs/pretrained_models.html to retrieve semantic-relevant news texts under cosine distance.

3.3 User Graph Encoding Channel

In this section, we will explain the user graph construction and graph context learning.

3.3.1 User Graph Construction

Motivated by Hu et al. (2020a) and Wu et al. (2021), we model user history with graph structure to encode multi-levels of user interests. We build a heterogeneous user graph G_u containing **news nodes** and **topic nodes**: (1) For a user's historical clicked news $H_u = [n_1, n_2, ..., n_{|H|}]$, we treat it as a set of historical news nodes for news-level user interest representation. (2) For the clicked news n_j, it is pertaining to a certain news topic[4] $T(i)$. We treat the clicked news topics as historical topic nodes for topic-level user interest representation.

To capture the interaction among news and topics, we introduce three types of edges:

News-News Edge. News nodes with the same topic category (e.g., *Sports*) are fully connected. In this way, we can capture the relatedness among clicked news with news-level interaction.

News-Topic Edge. We model the interaction between clicked news and topics by connecting news nodes to their corresponding topic nodes.

Topic-Topic Edge. Topic nodes are fully connected. In this way, we can capture the overall user interests with topic-level interaction.

3.3.2 User Graph Context Extraction

Given the user history $H_u = [n_1, n_2, ..., n_{|H|}]$, we employ the semantic news encoder (described in Section 3.1) to learn the historical news embeddings $h_u^n = [h_{u,1}^n, h_{u,2}^n, ..., h_{u,|H|}^n] \in \mathbb{R}^{|H| \times d}$. Given $|T(\cdot)|$ topics indicated by the clicked news, the topic nodes are embedded into learnable embeddings $h_u^t = [h_{u,1}^t, h_{u,2}^t, ..., h_{u,|T(\cdot)|}^t] \in \mathbb{R}^{|T(\cdot)| \times d}$. The user graph embeddings are as $h_u = [h_u^n, h_u^t]$.

Following Qi et al. (2021c), we extract the graph context $c_u \in \mathbb{R}^d$ in a hierarchical way. First, we employ an attention module to learn the topic representation $\tilde{h}_{u,T(i)} \in \mathbb{R}^d$ of the topic $T(i)$. The topic-attention module regards the news graph context c_n as the query and the news embeddings $\{h_{u,j}^n\}_{n_j \in T(i)}$ of topic $T(i)$ as the key-value pairs.

$$\tilde{h}_{u,T(i)} = \text{Attention}\big(c_n, \{h_{u,j}^n\}, \{h_{u,j}^n\}\big) \quad (7)$$

Then, we employ another attention module to extract the user graph context $c_u \in \mathbb{R}^d$. The user-attention module regards the news graph context c_n as the query and the learned topic representations $\{\tilde{h}_{u,T(i)}\}_{i=1}^{|T(\cdot)|}$ as the key-value pairs.

$$c_u = \text{Attention}\big(c_n, \{\tilde{h}_{u,T(i)}\}, \{\tilde{h}_{u,T(i)}\}\big) \quad (8)$$

$\text{Attention}(\cdot,\cdot,\cdot)$ in Eq. (7) and (8) denotes the standard attention module with Query/Key/Value. We implement $\text{Attention}(\cdot,\cdot,\cdot)$ as scaled dot-product attention (Vaswani et al., 2017) in our experiments.

[4]For example, in the *MIND* (Wu et al., 2020) dataset, each news has a topic category (e.g., *Sports* and *Entertainment*).

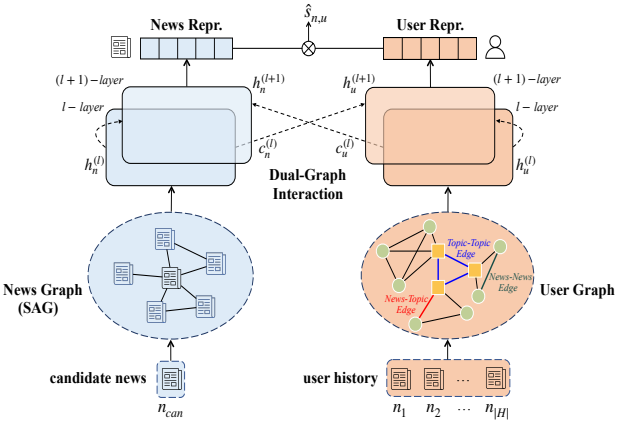


Figure 2: The overall architecture of DIGAT.

3.4 Dual-Graph Interaction

In news graph G_n, node embeddings $\{h_{n,i}\}_{i=0}^{|G_n|}$ contain the information of augmented candidate news semantics. In user graph G_u, node embeddings $\{h_{u,i}\}_{i=0}^{|G_u|}$ contain the information of user history. We learn informative news and user graph embeddings by aggregating neighboring node information with stacked graph attention layers (Veličković et al., 2018). Most importantly, our dual-graph interaction model aims at facilitating effective feature interaction between the news and user graphs. By effective dual-graph feature interaction, accurate news-user representation matching can be achieved. In the dual-graph interaction, the $(l+1)$-layer news node embeddings $h_n^{(l+1)}$ is updated based on the l-layer news node embeddings $h_n^{(l)}$ and user graph context $c_u^{(l)}$ jointly (vice versa to update the user node embeddings $h_u^{(l+1)}$), as illustrated in Figure 2.

We illustrate the news node embedding update process for example. We first perform a linear transformation on the l-layer news node embedding $h_{n,i}^{(l)}$ to derive higher-level graph features $\hat{h}_{n,i}$:

$$\hat{h}_{n,i} = \hat{\mathbf{W}}_n^l h_{n,i}^{(l)} + \hat{\mathbf{b}}_n^l, \quad (9)$$

where $\hat{\mathbf{W}}_n^l \in \mathbb{R}^{d \times d}$ and $\hat{\mathbf{b}}_n^l \in \mathbb{R}^d$ are learnable.

In order to learn news node embeddings interacting with user graph, we integrate the user graph

context $c_u^{(l)}$ into news graph attention computation. For news node i and node $j \in \mathcal{N}_i^n$ (where $\mathcal{N}_i^n$ is the neighborhood of node i), we incorporate user graph context $c_u^{(l)}$ into computing the attention key vector $K_{i,j}$. We use a feed-forward network $\text{FFN}_n^{(l)}$ to compute $K_{i,j}$ based on the fused information of $c_u^{(l)}$, $\hat{h}_{n,i}$ and $\hat{h}_{n,j}$. The news graph attention coefficient $\alpha_{i,j}$ is computed aware of user graph context.

$$K_{i,j} = \text{FFN}_n^{(l)}\Big(\big[c_u^{(l)}; \hat{h}_{n,i}; \hat{h}_{n,j}\big]\Big) \quad (10)$$

$$\alpha_{i,j} = \frac{\exp\Big(\text{LeakyReLU}\big(\mathbf{a}_n^T K_{i,j}\big)\Big)}{\sum_{k \in \mathcal{N}_i^n} \exp\Big(\text{LeakyReLU}\big(\mathbf{a}_n^T K_{i,k}\big)\Big)} \quad (11)$$

, where $\mathbf{a}_n^T$ is a learnable attention weight vector. Finally, we aggregate the neighboring node embeddings with attention coefficient $\alpha_{i,j}$, followed by ReLU activation. Residual connection is applied to mitigate gradient vanishing in deep graph layers.

$$h_{n,i}^{(l+1)} = \text{ReLU}\Big(\sum_{j \in \mathcal{N}_i^n} \alpha_{i,j} \hat{h}_{n,j}\Big) + h_{n,i}^{(l)} \quad (12)$$

The news and user graph contexts $c_n^{(l)}$ and $c_u^{(l)}$ are extracted from the l-layer graph node embeddings as described in Section 3.2.2 and 3.3.2. We summarize Eq. (9) to (12) as the news node embedding update function $\Phi_n^{(l)}$:

$$h_{n,i}^{(l+1)} = \Phi_n^{(l)}\Big(c_u^{(l)}, h_{n,i}^{(l)}, \big\{h_{n,j}^{(l)}\big\}_{j \in \mathcal{N}_i^n}\Big) \quad (13)$$

Similarly, the user node embedding update function is formulated as $\Phi_u^{(l)}$:

$$h_{u,i}^{(l+1)} = \Phi_u^{(l)}\Big(c_n^{(l)}, h_{u,i}^{(l)}, \big\{h_{u,j}^{(l)}\big\}_{j \in \mathcal{N}_i^u}\Big) \quad (14)$$

The dual-graph interaction can be viewed as an iterative process that performs (1) user graph context-aware attention to update news node embeddings and (2) news graph context-aware attention to update user node embeddings. We model the dual interaction with L stacked layers. The final layers of news and user graph contexts c_n^L and c_u^L are adopted as news and user graph representations r_n and r_u, which refine the news and user graph information with deep feature interaction. Algorithm 1 illustrates the dual-graph interaction process.

3.5 Click Prediction and Model Training

With the news and user graph representations r_n and r_u, our model aims to predict the matching score $\hat{s}_{n,u}$, which signals how likely user u will click news n. Motivated by An et al. (2019), we compute the news-user representation matching score by dot product as $\hat{s}_{n,u} = r_n^T r_u$.

Algorithm 1 News-User Graph Interaction

Input: news node embeddings $h_n^0 = \{h_{n,i}^0\}_{i=0}^{|G^n|}$, user node embeddings $h_u^0 = \{h_{u,i}^0\}_{i=0}^{|G^u|}$, number of dual-graph layers L.
Output: news graph representation r_n and user graph representation r_u.

```
Initialize c_n^0 from h_n^0 with Eq. (3) - (6).
Initialize c_u^0 from h_u^0 with Eq. (7) - (8).
for l = 0, 1, ..., L − 1 do
    Update the (l + 1)-layer news node embeddings h_n^(l+1) with Eq. (13).
    Update the (l + 1)-layer user node embeddings h_u^(l+1) with Eq. (14).
    Update the (l + 1)-layer news graph context c_n^(l+1) with Eq. (3) - (6).
    Update the (l + 1)-layer user graph context c_u^(l+1) with Eq. (7) - (8).
end for
r_n = c_n^L and r_u = c_u^L.
return r_n, r_u
```

Following Wu et al. (2019b,d), we use negative sampling approach to train our model. For the user behavior that user u had clicked news n_i, we compute the click matching score as $\hat{s}_i^+$ for n_i and u. Besides, we randomly sample S non-clicked news $[n_1, n_2, ..., n_S]$ from the user's behavior log and compute the negative matching scores as $[\hat{s}_{i,1}^-, \hat{s}_{i,2}^-, ..., \hat{s}_{i,S}^-]$. We optimize the NCE loss $\mathcal{L}$ over the training dataset $\mathcal{D}$ in model training.

$$\mathcal{L} = -\sum_{i=1}^{|\mathcal{D}|} log \frac{\exp(\hat{s}_i^+)}{\exp(\hat{s}_i^+) + \sum_{j=1}^{S} \exp(\hat{s}_{i,j}^-)} \quad (15)$$

4 Experiments

4.1 Dataset and Experiment Settings

We conduct experiments on the real-world benchmark dataset *MIND* (Wu et al., 2020). *MIND* is constructed from anonymized user behavior logs of Microsoft News with two versions of *MIND-large* and *MIND-small*. *MIND-large* contains 1 million anonymized users with user-click impression logs of 6 weeks from October 12 to November 22, 2019. The training and dev sets contain the impression logs of the first 5 weeks, and the last week impression logs are reserved for test. *MIND-small* consists

#	Method	*MIND-small* AUC	MRR	nDCG@5	nDCG@10	*MIND-large* AUC	MRR	nDCG@5	nDCG@10
1	GRU	61.51	27.46	30.11	36.61	65.42	31.24	33.76	39.47
2	DKN	62.90	28.37	30.99	37.41	64.07	30.42	32.92	38.66
3	NAML	66.12	31.53	34.88	41.09	66.46	32.75	35.66	41.40
4	NPA	64.65	30.01	33.14	39.47	65.92	32.07	34.72	40.37
5	LSTUR	65.87	30.78	33.95	40.15	67.08	32.36	35.15	40.93
6	NRMS	65.63	30.96	34.13	40.52	67.66	33.25	36.28	41.98
7	FIM	65.34	30.64	33.61	40.16	67.87	33.46	36.53	42.21
8	HieRec	67.83	32.78	36.31	42.49	69.03	33.89	37.08	43.01
9	GERL	65.27	30.10	32.93	39.48	68.10	33.41	36.34	42.03
10	GNewsRec	65.54	30.27	33.29	39.80	68.15	33.45	36.43	42.10
11	User-as-Graph†	–	–	–	–	69.23	34.14	37.21	43.04
	DIGAT	68.39	33.08	36.71	42.92	69.96	34.78	38.05	43.76

Table 1: Evaluation results of all methods. Experiments of baseline #1 to #10 and DIGAT are conducted 10 times on *MIND-small* and 5 times on *MIND-large*, respectively. We report the average performance. †Results of *User-as-Graph* are directly copied from the previous works (Wu et al., 2021).

of 50000 users, which are randomly sampled from *MIND-large* with the impression logs.

Following previous works (Wang et al., 2020; Qi et al., 2021c), we use news titles with the maximum length of 32 words for news textual encoding. The user history includes 50 news items they have recently clicked. The news word embeddings are 300-dimensional and initialized from the pretrained Glove embeddings (Pennington et al., 2014). Following An et al. (2019), we set the number of negative news samples S to be 4. For our model parameters, the number of CNN feature maps d is set as 400. The number of neighboring nodes M and hops K are 5 and 2, respectively. We set the number of dual-graph interaction layers as $L = 4$. We use Adam optimizer (Kingma and Ba, 2015) with the learning rate of 1e-4 to train our model. Following Wu et al. (2020), we employ the recommendation ranking metrics AUC, MRR, nDCG@5 and nDCG@10 to evaluate model performance.

4.2 Compared Methods

We compare our model with the state-of-the-art news recommendation methods: (1) *GRU* (Okura et al., 2017), learning user representations from a sequence of clicked news with a GRU network; (2) *DKN* (Wang et al., 2018), using a knowledge-aware CNN to learn news representations from both news texts and knowledge entities; (3) *NAML* (Wu et al., 2019a), learning news representations from news titles, bodies, categories and subcategories with multi-view attention networks; (4) *NPA* (Wu et al., 2019b), encoding news and user representations with personalized attention networks; (5) *LSTUR* (An et al., 2019), jointly modeling long-term user embeddings and short-term user interests learned by a GRU network; (6) *NRMS* (Wu et al., 2019d) encoding informative news and user representations with multi-head self-attention networks; (7) *FIM* (Wang et al., 2020), encoding news content with dilated convolutional networks and modeling user interest matching with 3-D convolutional networks; (8) *HieRec* (Qi et al., 2021c), modeling user interests in a three-level hierarchy and performing multi-grained matching between candidate news and hierarchical user interest representations.

We also compare our model with competitive graph-based methods: (9) *GERL* (Ge et al., 2020), modeling the news-user relatedness with a bipartite graph, which enhances news and user representations by aggregating neighboring node information; (10) *GNewsRec* (Hu et al., 2020a), using graph neural networks (GNN) (Hamilton et al., 2017) to encode long-term user interests from a user-news-topic graph; (11) *User-as-Graph* (Wu et al., 2021), utilizing a heterogeneous graph pooling method to extract user representations from personalized heterogeneous behavior graphs.

4.3 Main Experiment Results

In Table 1, we present the main experiment results. We can observe that *DIGAT* significantly outperforms the general two-tower encoder methods (i.e., methods #1 to #8) on the both datasets. This is because even though some baselines use topic categories or knowledge entities to enrich news information (e.g., *HieRec* learns news representations from both news texts and knowledge entities), the information entailed in single candidate news may be still insufficient. In contrast, *DIGAT* can sub-

	AUC	MRR	nDCG@5	nDCG@10
w/o SA	67.57	32.38	35.81	42.10
TF-IDF SA	67.76	32.61	36.14	42.39
SA-Seq	68.05	32.75	36.41	42.62
DIGAT	68.39	33.08	36.71	42.92

Table 2: Experiment results of SAG modeling variants.

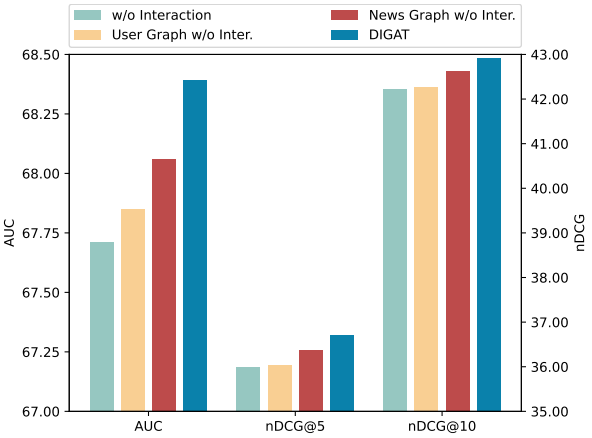


Figure 3: Ablation results on dual-graph interaction.

stantially enrich the semantic information of the single candidate news by SAG modeling, which provides more accurate candidate news signals to match user interests. Besides, *DIGAT* significantly outperforms three graph-based baselines. We find that *GERL* is hard to model news-user interaction in test data, as most candidate news items in test data are fresh and have no click-interaction with users. Differently, *DIGAT* models news and users with dual graph channels instead of a joint bipartite graph, which circumvents this *cold news* issue. Compared to *GNewsRec* and *User-as-Graph*, *DIGAT* performs more effective feature interaction between the news and user graphs, which can enhance more accurate news-user representation matching.

4.4 Effectiveness of SAG Modeling

We examine the effectiveness of SAG modeling from three perspectives: (1) To examine the effectiveness of semantic-augmentation (SA) strategy, we remove SAG from our model and instead learn single candidate news representation (**w/o SA**). (2) To inspect the function of PLM $\phi(\cdot)$ in SAG construction (see Section 3.2.1), we conduct controlled experiments by replacing $\phi(\cdot)$ with TF-IDF feature extractor (**TF-IDF SA**). (3) To examine the effectiveness of graph-based SA, we conduct controlled experiments by organizing the semantic-relevant news in a sequential form and extracting the news sequence context similar to Eq. (3)-(6) (**SA-Seq**). The experiments in this section and following sections are conducted on *MIND-small*.

Table 2 shows the experiment results. We can see that abandoning the SA strategy (**w/o SA**) leads to the largest performance drop, as **TF-IDF SA** and **SA-Seq** also yield better performance than **w/o SA**. This validates the effectiveness of SA strategy to enrich candidate news semantics and further enhance news recommendation. **TF-IDF SA** underperforms our original approach significantly. We infer that the TF-IDF features can only evaluate news similarity at the syntactic level, which may not be able to accurately retrieve semantic-relevant news for SAG construction. In contrast, PLM can accurately evaluate news similarity at the semantic level and help retrieve more relevant news to enhance SAG modeling. Besides, **SA-Seq** is suboptimal compared to the original graph-based SA. This is because the graph-based SA method can accurately model the relatedness among the candidate news and semantic-relevant news with multi-neighbor and multi-hop graph structure, which further improves the effectiveness of SA strategy.

4.5 Ablation Study on Graph Interaction

To examine the effectiveness of dual-graph interaction, we design the following ablation experiments: (1) **w/o Interaction**. We employ the vanilla graph attention networks (GAT) (Veličković et al., 2018) to learn news and user graph embeddings, respectively, without interaction between dual graphs. (2) **News Graph w/o Inter**. The news graph embedding update layers are replaced with vanilla GAT layers. Concretely, Eq. (13) is modified into $h_{n,i}^{(l+1)} = \bar{\Phi}_n^{(l)}\big(h_{n,i}^{(l)}, \{h_{n,j}^{(l)}\}_{j\in\mathcal{N}_i^n}\big)$, where $\bar{\Phi}_n^{(l)}$ is the standard GAT graph embedding update function without feature interaction with user graph context. (3) **User Graph w/o Inter**. Similar to (2), we replace the user graph embedding update layers with vanilla GAT layers.

Figure 3 shows the performance of the ablation models. We can see that **w/o Interaction** underperforms the other three models with graph interaction modeling. It indicates that feature interaction between candidate news and users is necessary to enhance news recommendation. Besides, we can observe that removing user graph interaction (**User Graph w/o Inter**) leads to more performance drop than **News Graph w/o Inter**, which implies that user graph interaction may contribute more to our model. Moreover, DIGAT surpasses the two single

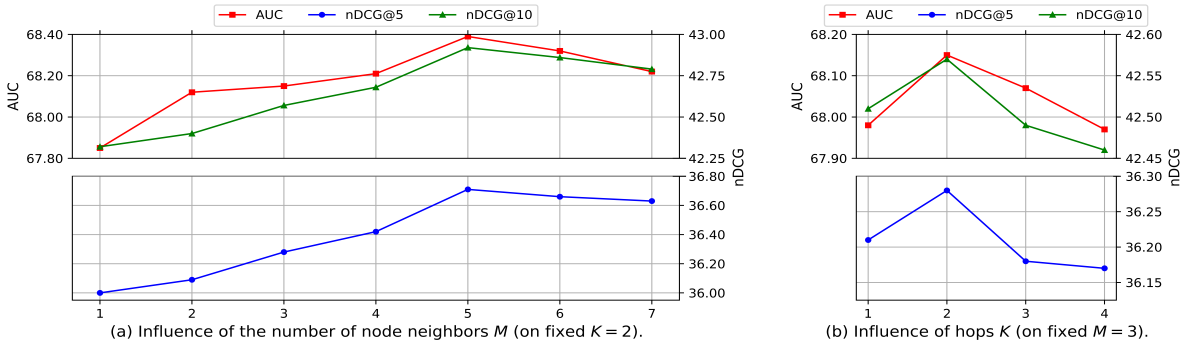


Figure 4: DIGAT performance with different M and K settings of SAG.

graph interaction ablations by a significant margin, validating the effectiveness of modeling dual-graph interaction in a deep and iterative manner.

4.6 Analysis on SAG Parameters

We investigate two key parameters of SAG, i.e., the number of node neighbors M and hops K. Figure 4 shows the influence of different M and K settings.

As shown in Figure 4(a), DIGAT performance continues rising as M increases from 1 to 5. This is because with more semantic-relevant news incorporated, SAG can leverage more fine-grained semantic information to enhance the candidate news representations. It can be observed that the model performance slightly declines when $M > 5$. The reason could be twofold. First, as the scale of SAG grows larger, it becomes more challenging for our model to distill the global graph context of SAG (see Section 3.2.2). Second, as M becomes too large, it is inevitable to retrieve more noisy news in the SAG construction process, which may hurt the SAG modeling. From Figure 4(b), we find that $K = 2$ is the optimal hop setting. This may be because two hops of SAG can heuristically capture more useful semantic-relevant news information than simple one-hop modeling, while higher-hop extension may introduce too much irrelevant news and interfere with accurate semantic augmentation for candidate news. In general, we select $M = 5$ and $K = 2$ on our SAG construction[5].

4.7 The Number of Dual-Graph Layers

We study the influence of the number of dual-graph layers L on DIGAT. The results are presented in Figure 5. We can see that the model performance first keeps increasing and reaches a peak at $L = 4$.

[5]The SAG construction ($M = 5$ and $K = 2$) on *MIND-large* can be finished in 15 minutes on Intel(R) Xeon(R) Gold 6226R CPU @ 2.90 GHz with Nvidia V100 GPU.

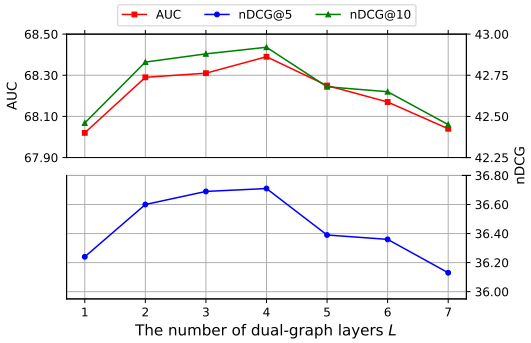


Figure 5: DIGAT performance with different numbers of dual-graph layers L.

It suggests that deep feature interaction between news and user graphs is useful to improve recommendation performance, as it can model news and user representation matching in a more fine-grained way. We can also observe that further increasing L hurts the model performance. It may be caused by the unstable gradient in training the deep dual-graph architecture, as we empirically find that gradient clipping (Pascanu et al., 2013) is indispensable to successfully train DIGAT when the dual-graph layers are too deep (i.e., $L = 6, 7$).

5 Conclusion

In this work, we present a dual-graph interaction framework for news recommendation. In our approach, a graph enhanced semantic-augmentation strategy is employed to enrich the semantic information of candidate news. Moreover, we design a dual-graph interaction mechanism to achieve effective feature interaction between news and user graphs, facilitating more accurate news and user representation matching. Our approach advances the state-of-the-art news recommendation methods on the MIND benchmark dataset. Extensive experiments and further analysis validate that SAG modeling and dual-graph interaction can effectively improve news recommendation performance.

References

M. Tarik Altuncu, Sophia N. Yaliraki, and Mauricio Barahona. 2018. Content-driven, unsupervised clustering of news articles through multiscale graph partitioning. In *arXiv preprint arXiv:1808.01175*.

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*, pages 336–345, Florence, Italy. Association for Computational Linguistics.

Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 27–34. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph enhanced representation learning for news recommendation. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2863–2869. ACM / IW3C2.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2020a. Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management*, 57:102142.

Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020b. Graph neural news recommendation with unsupervised preference disentanglement. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4255–4264, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1933–1942, New York, NY, USA. Association for Computing Machinery.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA. PMLR.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021a. Personalized news recommendation with knowledge-aware interactive matching. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 61–70. ACM.

Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021b. PP-rec: News recommendation with personalized user interest and time-aware news popularity. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5457–5467, Online. Association for Computational Linguistics.

Tao Qi, Fangzhao Wu, Chuhan Wu, Peiru Yang, Yang Yu, Xing Xie, and Yongfeng Huang. 2021c. HieRec: Hierarchical user interest modeling for personalized news recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5446–5456, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages

3982–3992, Hong Kong, China. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*.

Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 836–845, Online. Association for Computational Linguistics.

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*, WWW '18, page 1835–1844, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019a. Neural news recommendation with attentive multi-view learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3863–3869. International Joint Conferences on Artificial Intelligence Organization.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019b. Npa: Neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 2576–2584, New York, NY, USA. Association for Computing Machinery.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Yongfeng Huang, and Xing Xie. 2019c. Neural news recommendation with topic-aware news representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1154–1159, Florence, Italy. Association for Computational Linguistics.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019d. Neural news recommendation with multi-head self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6389–6394, Hong Kong, China. Association for Computational Linguistics.

Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2021. User-as-graph: User modeling with heterogeneous graph pooling for news recommendation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1624–1630. International Joint Conferences on Artificial Intelligence Organization. Main Track.

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*, pages 3597–3606, Online. Association for Computational Linguistics.

Jingwei Yi, Fangzhao Wu, Chuhan Wu, Ruixuan Liu, Guangzhong Sun, and Xing Xie. 2021. Efficient-FedRec: Efficient federated learning framework for privacy-preserving news recommendation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. Dan: Deep attention neural network for news recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5973–5980.

Algorithm 2 SAG Construction Procedure

Input: candidate news n_0, training news corpus $\{N_T\}$, node neighbors M and hops K.
Output: semantic-augmented graph G_n

1: Regard n_0 as the root node v_0 of SAG.
2: Initialize graph node set $V \leftarrow \{v_0\}$ and edge set $E \leftarrow \{\}$. Define parent node set $P \leftarrow \{v_0\}$ and node-hop counter $\text{hop}[v_0] = 0$.
// Graph extension process
3: **while** $P \neq \emptyset$ **do**
4: Pop a node v_i from P, then $P = P \setminus \{v_i\}$
5: Retrieve M news $\{n_j\}_{j=1}^{M}$ from the news corpus $\{N_T\}$ with the M highest semantic similarity scores $\{s_{i,j}\}_{j=1}^{M}$ as nodes $\{v_j\}_{j=1}^{M}$
6: **for** $j = 1, 2, ..., M$ **do**
7: **if** $v_j \notin V$ **then**
8: $V = V \cup \{v_j\}$
9: $\text{hop}[v_j] = \text{hop}[v_i] + 1$
10: **if** $\text{hop}[v_j] < K$ **then**
11: $P = P \cup \{v_j\}$
12: **end if**
13: **end if**
14: **if** edge $e_{i,j} \notin E$ **then**
15: $E = E \cup \{e_{i,j}\}$
16: **end if**
17: **end for**
18: **end while**
19: $G_n = \{V, E\}$.
20: **return** G_n

A Semantic-augmented Graph Construction and Examples

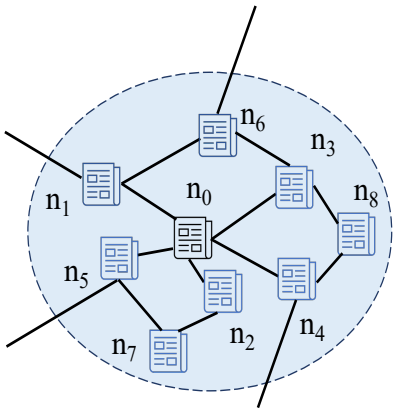
Algorithm 2 illustrates the procedure of semantic-augmented graph (SAG) construction. First of all, the SAG G_n is initialized from the root node v_0, which represents the original candidate news n_0.

The graph construction is performed by repeatedly extending semantic-relevant neighboring news nodes to existing nodes in G_n. In the **graph extension process** (line 3 to 18 in Algorithm 2), for an existing node v_i (corresponding to news n_i) in G_n, we retrieve M news documents $\{n_j\}_{j=1}^{M}$ from the training news corpus[6] $\{N_T\}$ with the M highest similarity scores $\{s_{i,j}\}_{j=1}^{M}$. The similarity score $s_{i,j}$ of news n_i and n_j is evaluated by a PLM $\phi(\cdot)$ with Eq. (2). We treat the retrieved news $\{n_j\}_{j=1}^{M}$ as news nodes $\{v_j\}_{j=1}^{M}$. For each node v_j, we extend it to G_n as a neighboring node of v_i by adding bidirectional edge $e_{i,j}$ between v_i and v_j. To heuristically explore higher-order semantic-relevant news, news nodes in SAG are extended from the root node v_0 within K hops at most.

SAG Examples. Figure 6 demonstrates an example of SAG instance for the candidate news n_0 "*Should the NFL be able to fine players for criticizing officiating*". Interestingly, from Figure 6(b), we can see that there are many similar news items in SAG regarding a specific event or person (i.e., "*NFL*" and "*fine players*") from different perspectives. These semantic-relevant news documents can be finely retrieved with the help of PLM and substantially enrich the semantic information of the original candidate news.

News Clustering Phenomenon. From the example SAG shown in Figure 6(a), we find that there exist many cyclic sub-graphs, revealing the news clustering phenomenon in the semantic space. This cyclic graph structure indicates the presence of similar news clusters at different levels of granularity, coinciding with the previous research (Altuncu et al., 2018). It also justifies the motivation of our work. By augmenting the semantic-relevant information, SAG is capable of capturing robust and informative representations of the candidate news.

[6] In our experiments, we utilize news in the *train/news.tsv* data file of *MIND* dataset to construct the news corpus.



(a)

News	Title	Neighbors
n_0	*Should the NFL be able to fine players for criticizing officiating?*	[1,2,3,4,5]
n_1	*NFL sending message with multiple fines for criticizing referees*	[0,6]
n_2	*NFL cracks down on criticizing refs with fines for Baker Mayfield, Clay Matthews*	[0,7]
n_3	*NFL cracks down on internal dissent over officiating*	[0,6,8]
n_4	*NFL fines Baker Mayfield for stating the obvious*	[0,8]
n_5	*Biggest blown call of season may prove NFL officials are wrecking new pass interference rule*	[0,7]
n_6	*Mayfield fined after comments on officiating following loss to seahawks*	[1,3]
. . . . .		

(b)

Figure 6: An example of SAG ($M = 5$ and $K = 2$) constructed from news n_0 in *MIND-large* (news ID: N124534): (a) A subgraph of the example SAG including root node n_0 and semantic-relevant news node n_i ($i = 1, 2, ..., 8$); (b) News in SAG and the corresponding title texts. For brevity, we only present an SAG subgraph of news nodes.