Graph Enhanced Representation Learning for News Recommendation

Suyu Ge

Chuhan Wu

Fangzhao Wu

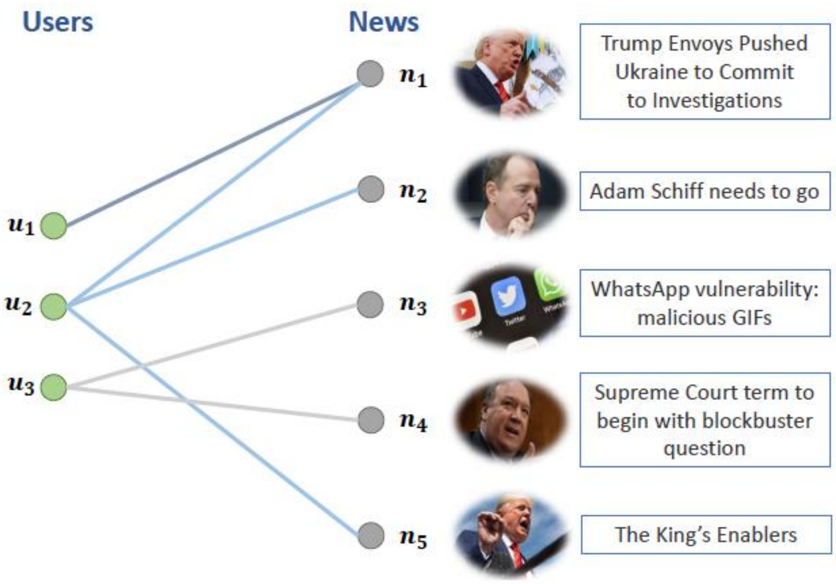
Tao Qi

Yongfeng Huang

Introduction:

General Personalized Recommendation(ID-based recommendation) → Data Sparsity problem → cold-start problem

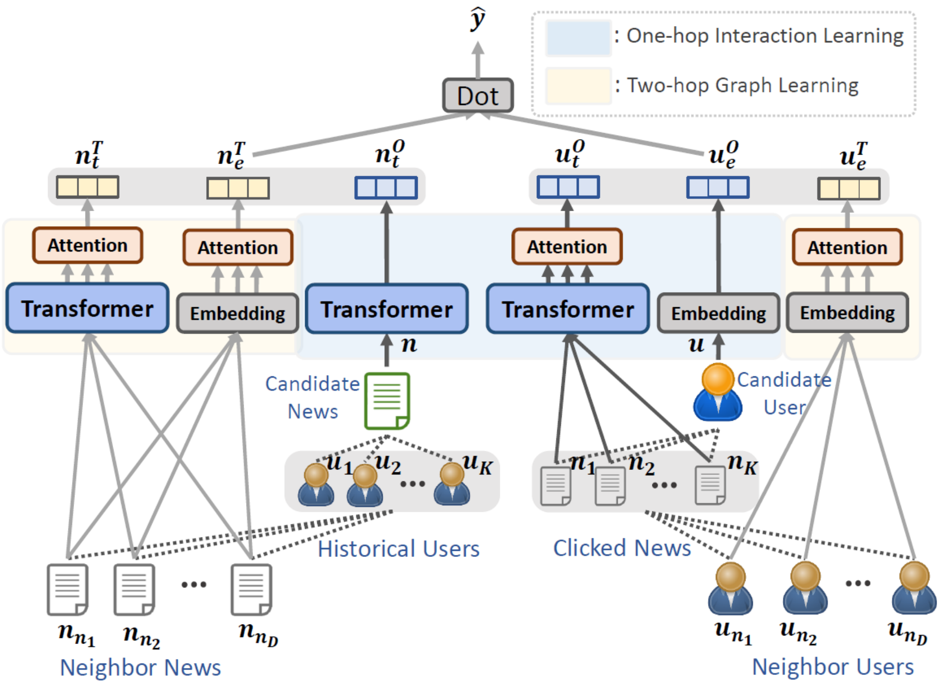
Recently neural network recommendation(DKN, NPA) → which depends on clicked news → clicked news are sparse → cold-start problem


Users
News
u_1
u_2
u_3
n_1
n_2
n_3
n_4
n_5
Trump Envoys Pushed Ukraine to Commit to Investigations
Adam Schiff needs to go
WhatsApp vulnerability: malicious GIFs
Supreme Court term to begin with blockbuster question
The King's Enablers

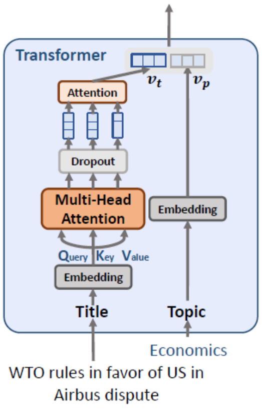
Three observations for this work:

1. A bipartite graph can be built from user-news interactions.
2. News representations can be enriched by considering neighbor news in the graph.
3. User representations can be enriched by neighbor users.

APPROACH:



1. Transformer for Context Understanding



The representation of the i-thword learned

by the k-th attention head is computed as:

$$\alpha^k_{i,j} = \frac{\exp(\mathbf{e}_i^T \mathbf{W}_s^k \mathbf{e}_j)}{\sum_{m=1}^{M} \exp(\mathbf{e}_i^T \mathbf{W}_s^k \mathbf{e}_m)},$$

$$\mathbf{h}_i^k = \mathbf{W}_v^k (\Sigma_{j=1}^{M} \alpha^k_{i,j} \mathbf{e}_j),$$

Additive attention: $$\beta_i^w = \frac{\exp(\mathbf{q}_w^T \tanh(\mathbf{U}_w \times \mathbf{h}_i + \mathbf{u}_w))}{\sum_{j=1}^{M} \exp(\mathbf{q}_w^T \tanh(\mathbf{U}_w \times \mathbf{h}_j + \mathbf{u}_w))},$$

2. One-hop Interaction Learning

(1) Candidate news semantic representations:

Using transformer to learn candidate news.

(2) Target user semantic representations:

Additive attention.

(3) Target user ID representations.

ID embedding.

3. Two-hop Graph Learning

(1) Neighbor user ID representations:

ID embedding → Additive Attention

(2) Neighbor news ID representations:

Additive Attention

(3)Neighbor news semantic representations:

Transformer → Additive Attention

Graph Enhanced Representation Learning for News Recommendation

Suyu Ge
Tsinghua National Laboratory for Information Science and Technology
Tsinghua University
gesy17@mails.tsinghua.edu.cn

Chuhan Wu
Tsinghua National Laboratory for Information Science and Technology
Tsinghua University
wuchuhan15@gmail.com

Fangzhao Wu
Microsoft Research Asia
Beijing, China
wufangzhao@gmail.com

Tao Qi
Tsinghua National Laboratory for Information Science and Technology
Tsinghua University
qit16@mails.tsinghua.edu.cn

Yongfeng Huang
Tsinghua National Laboratory for Information Science and Technology
Tsinghua University
yfhuang@tsinghua.edu.cn

ABSTRACT

With the explosion of online news, personalized news recommendation becomes increasingly important for online news platforms to help their users find interesting information. Existing news recommendation methods achieve personalization by building accurate news representations from news content and user representations from their direct interactions with news (e.g., click), while ignoring the high-order relatedness between users and news. Here we propose a news recommendation method which can enhance the representation learning of users and news by modeling their relatedness in a graph setting. In our method, users and news are both viewed as nodes in a bipartite graph constructed from historical user click behaviors. For news representations, a transformer architecture is first exploited to build news semantic representations. Then we combine it with the information from neighbor news in the graph via a graph attention network. For user representations, we not only represent users from their historically clicked news, but also attentively incorporate the representations of their neighbor users in the graph. Improved performances on a large-scale real-world dataset validate the effectiveness of our proposed method.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Natural language processing**.

KEYWORDS

News Recommendation, Transformer, Graph Attention Network

ACM Reference Format:
Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph Enhanced Representation Learning for News Recommendation. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366423.3380050

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380050

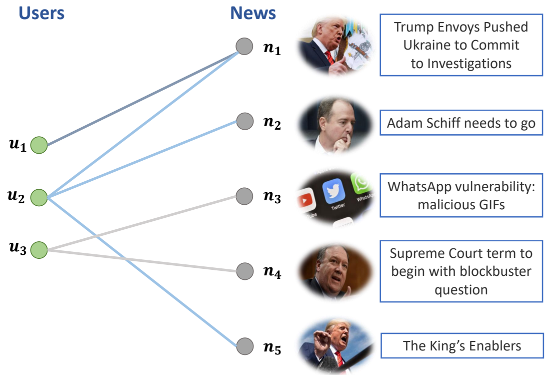

Users
News
u_1
u_2
u_3
n_1
n_2
n_3
n_4
n_5
Trump Envoys Pushed Ukraine to Commit to Investigations
Adam Schiff needs to go
WhatsApp vulnerability: malicious GIFs
Supreme Court term to begin with blockbuster question
The King's Enablers

Figure 1: A user-news bipartite graph.

1 INTRODUCTION

Both the overwhelming number of newly-sprung news and huge volumes of online news consumption pose challenges to online news aggregating platforms. Thus, how to target different users' news reading interests and avoid showcasing excessive irrelevant news becomes an important problem for these platforms [14, 19]. A possible solution is personalized news recommendation, which depicts user interests from previous user-news interactions [2, 10]. However, unlike general personalized recommendation methods, news recommendation is unique from certain aspects. The fast iteration speed of online news makes traditional ID-based recommendation methods such as collaborative filtering (CF) suffer from data sparsity problem [4]. Meanwhile, rich semantic information in news texts distinguishes itself from recommendation in other domains (e.g., music, fashion and food). Therefore, a precise understanding of textual content is also vital for news recommendation.

Existing news recommendation methods achieve personalized news ranking by building accurate news and user representations. They usually build news representations from news content [2, 11, 32, 38]. Based on that, user representations are constructed from their click behaviors, e.g., the aggregation of their clicked news representations. For example, Wang et al. proposed DKN [25], which formed news representations from their titles via convolutional neural network (CNN). Then they utilized an attention network to select important clicked news for user representations. Wu et al. [31] further enhanced personalized news representations by incorporating user IDs as attention queries to select important words in news titles. When forming user representations, the same attention query was used to select important clicked news. Compared with traditional collaborative filtering methods [9, 13, 20], which suffer heavy cold-start problems [12], these methods gained a competitive edge by learning semantic news representations directly from news context. However, most of them build news representations only from news content and build user representations only from users' historically clicked news. When the news content such as titles are short and vague, and the historical behaviors of user are sparse, it is difficult for them to learn accurate news and user representations.

Our work is motivated by several observations. First, from user-news interactions, a bipartite graph can be established. Within this graph, both users and news can be viewed as nodes and interactions between them can be viewed as edges. Among them, some news are viewed by the same user, thus are defined as neighbor news. Similarly, specific users may share common clicked news, and are denoted as neighbor users. For example, in Figure 1, news n_1 and n_5 are neighbors because they are both clicked by user u_2. Meanwhile, u_1 and u_2 are neighbor users. Second, news representation may be enhanced by considering neighbor news in the graph. For example, neighbor news n_1 and n_5 both relates to politics. However, the expression "The King" in n_5 is vague without any external information. By linking it to news n_1, which is more detailed and explicit, we may infer that n_5 talks about president Trump. Thus, when forming news representation for n_5, n_1 may be modeled simultaneously as a form of complementary information. Third, neighbor users in the graph may share some similar news preferences. Incorporating such similarities may further enrich target user representations. As illustrated, u_1 and u_2 share common clicked news n_1, indicating that they may be both interested in political news. Nevertheless, it is challenging to form accurate user representation for u_1 since the click history of u_1 is very sparse. Thus, explicitly introducing information from u_2 may enrich the representation of u_1 and lead to better recommendation performances.

In this paper, we propose to incorporate the graph relatedness of users and news to enhance their representation learning for news recommendation. First, we utilize the transformer [24] to build news semantic representations from textual content. In this way, the multi-head self-attention network encodes word dependency in titles at both short and long distance. We also add topic embeddings of news since they may contain important information. Then we further enhance news representations by aggregating neighbor news via a graph attention network. To enrich neighbour news representations, we utilize both their semantic representations and ID embeddings. For user representations, besides attentively building user representations from their ID embeddings and historically clicked news, our approach also leverages graph information. We use the attention mechanism to aggregate the ID embeddings of neighbor users. Finally, recommendation is made by taking the dot product between user and news representations. We conduct extensive experiments on a large real-world dataset. The improved performances over a set of well-known baselines validate the effectiveness of our approach.

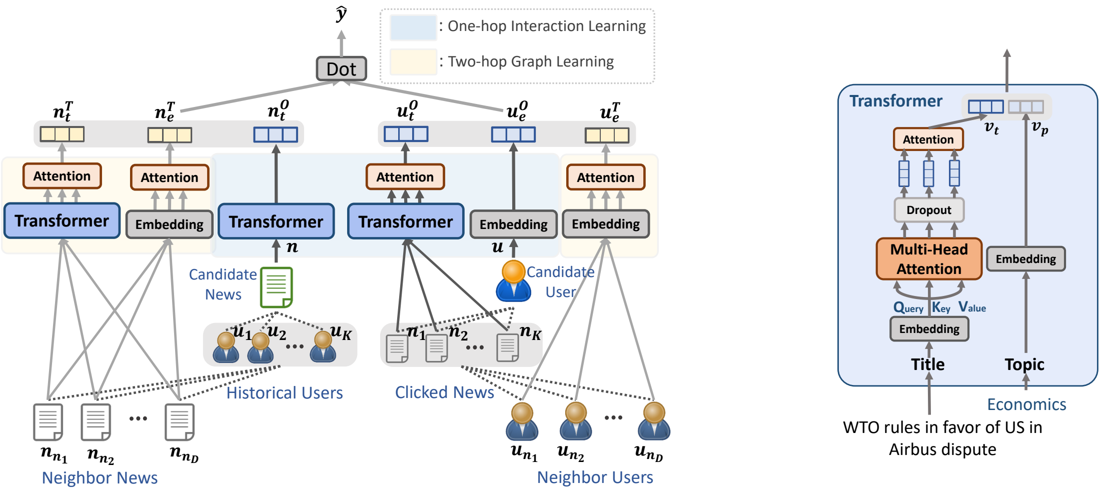
2 RELATED WORK

Neural news recommendation receives attention from both data mining and natural language processing fields [5, 29, 37]. Many previous works handle this problem by learning news and user representations from textual content [1, 30, 31, 38]. From such viewpoint, user representations are built upon clicked news representations using certain summation techniques (e.g., attentive aggregation or sequential encoding). For instance, Okura [17] incorporated denoising autoencoder to form news representations. Then they explored various types of recurrent networks to encode users. An et. al [1] attentively encoded news by combining title and topic information. They learned news representations via CNN and formed user representations from their clicked news via a gated recurrent unit (GRU) network. Zhu et. al. [38] exploited long short-term memory network (LSTM) to encode clicked news, then applied a single-directional attention network to select important click history for user representations. Though effective in extracting information from textual content, the works presented above neglect relatedness between neighbor users (or items) in the interaction graph. Different from their methods, our approach exploits both context meaning and neighbor relatedness in graph.

Recently, graph neural networks (GNN) have received wide attention, and a surge of attempts have been made to develop GNN architectures for recommender systems [5, 33, 35]. These models leverage both node attributes and graph structure by representing users and items using a combination of neighbor node embeddings [22]. For instance, Wang et. al. [27] combined knowledge graph (KG) with collaborative signals via a graph attention network, thus enhancing user and item representations with entity information in KG. Ying et. al. [35] introduced graph convolution to web-scale recommendation. Node representations of users and items were formed using visual and annotation features. In most works, representations are initially formed via node embedding, then optimized by receiving propagation signals from the graph [28, 33]. Although node embeddings are enhanced by adding item relation [34], visual features [35] or knowledge graphs [26], rich semantic information in the textual content may not be fully exploited. Different form their work, our approach learns the node embeddings of news directly from its textual content. We utilize the transformer architecture to model context dependency in news titles. Thus, our approach improves the node embedding by forming context-aware news representation.

3 OUR APPROACH

In this section, we will introduce our ***G**raph **E**nhanced **R**epresentation **L**earning* (**GERL**) approach illustrated in Figure 2, which consists of a *one-hop interaction learning* module and a *two-hop graph learning* module. The *one-hop interaction learning* module represents target user from historically clicked news and represents candidate news based on its textual content. The *two-hop graph learning* module learns neighbor embeddings of news and users using a graph attention network.



(a) Overview of the model.

(b) Transformer submodule.

Figure 2: An illustration of our proposed GERL approach. Dashed lines represent graph connectivity established from click behaviors, and solid lines represent the information flow among different modules.

3.1 Transformer for Context Understanding

Motivated by Vaswani et al. [24], we utilize the transformer to form accurate context representations from news titles and topics. News titles are usually clear and concise. Hence, to avoid the degradation of performance caused by excessive parameters, we simplify the transformer to single layer of multi-head attention.[1]

We then introduce the modified transformer from bottom to top. The bottom layer is the word embedding, which converts words in a news title into a sequence of low-dimensional embedding vectors. Denote a news title with M words as $[w_1, w_2, ..., w_M]$, through this layer it is converted into the embedded vector sequence $[\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_M]$.

The following layer is a word-level multi-head self-attention network. Interactions between words are important for learning news representations. For instance, in the title "Sparks gives Penny Toler a fire from the organization", the interaction between "Sparks" and "organization" helps understand the title. Moreover, a word may relate to more than one words in the title. For example, the word "Sparks" has interactions with both words "fire" and "organization". Thus, we employ the multi-head self-attention to form contextual word representations. The representation of the i^{th} word learned by the k^{th} attention head is computed as:

$$\alpha^k_{i,j} = \frac{\exp(\mathbf{e}_i^T \mathbf{W}_s^k \mathbf{e}_j)}{\sum_{m=1}^{M} \exp(\mathbf{e}_i^T \mathbf{W}_s^k \mathbf{e}_m)}, \qquad (1)$$

$$\mathbf{h}_i^k = \mathbf{W}_v^k(\Sigma_{j=1}^{M} \alpha^k_{i,j} \mathbf{e}_j),$$

where $\mathbf{W}_s^k$ and $\mathbf{W}_v^k$ are the projection matrices in the k^{th} self-attention head, and $\alpha^k_{i,j}$ indicates the relative importance of the relatedness between the i^{th} and j^{th} words. The multi-head representation $\mathbf{h}^i$ of the i^{th} word is the concatenation of the representations produced by N separate self-attention heads, i.e., $\mathbf{h}_i = [\mathbf{h}_i^1; \mathbf{h}_i^2; ...; \mathbf{h}_i^N]$. To mitigate overfitting, we add dropout [23] after the self-attention.

Next, we utilize an additive word attention network to model relative importance of different words and aggregate them into title representations. For instance, the word "fire" is more important than other words in the above example. The attention weight β_i^w of the i_{th} word is computed as:

$$\beta_i^w = \frac{\exp(\mathbf{q}_w^T \tanh(\mathbf{U}_w \times \mathbf{h}_i + \mathbf{u}_w))}{\sum_{j=1}^{M} \exp(\mathbf{q}_w^T \tanh(\mathbf{U}_w \times \mathbf{h}_j + \mathbf{u}_w))}, \qquad (2)$$

where $\mathbf{q}_w$, $\mathbf{U}_w$ and $\mathbf{u}_w$ are trainable parameters in the word attention network. The news title representation $\mathbf{v}_t$ is then calculated as: $\mathbf{v}_t = \Sigma_{i=1}^{M} \beta_i^w \mathbf{h}_i$.

Since topics of user clicked news may also reveal their preferences, we model news topics via an embedding matrix. Denote the output of this embedding matrix as $\mathbf{v}_p$, then the final representation of the news is the concatenation of the title vector and the topic vector, i.e., $\mathbf{v} = [\mathbf{v}_t; \mathbf{v}_p]$.

3.2 One-hop Interaction Learning

The *one-hop interaction learning* module learns candidate news and click behaviors of target users. More specifically, it can be decomposed into three parts: (1) Candidate news semantic representations; (2) Target user semantic representations; (3) Target user ID representations.

Candidate News Semantic Representations. Since understanding the content of candidate news is crucial for recommendation, we propose to utilize the transformer to form accurate

[1] We also tried the original transformer architecture but the performance is sub-optimal.

representation of it. Given the candidate news n, the one-hop (denoted as superscript ^{O}) output of the transformer module (denoted as subscript $_t$) is $\mathbf{n}_t^O$.

Target User Semantic Representations. The news reading preference of a user can be clearly revealed by their clicked news. Thus, we propose to model user representations from the content of their clicked news. Besides, different news may have varied importance for modeling user interests. For example, the news "crazy storms hit Los Angeles" is less important than the news "6 most popular music dramas" in modeling user interests. Thus, we apply an additive attention mechanism to aggregate clicked news vectors for user representations. Given a target user u and a total number of K clicked news $[n_1, n_2, ..., n_K]$, we first get their transformer encoded outputs $[\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_K]$. Then the attention weight β_i^n of the i^{th} clicked news is calculated as:

$$\beta_i^n = \frac{\exp(\mathbf{q}_n^T \tanh(\mathbf{U}_n \times \mathbf{v}_i + \mathbf{u}_n))}{\Sigma_{q=1}^K \exp(\mathbf{q}_n^T \tanh(\mathbf{U}_n \times \mathbf{v}_q + \mathbf{u}_n))}, \tag{3}$$

where $\mathbf{q}_n$, $\mathbf{U}_n$ and $\mathbf{u}_n$ are the trainable parameters of the news attention network. The one-hop user semantic representation $\mathbf{u}_t^O$ is then calculated as: $\mathbf{u}_t^O = \Sigma_{i=1}^K \beta_i^n \mathbf{v}_i$.

Target User ID Representations. Since user IDs represent each user uniquely, we incorporate them as latent representations of user interests [15, 16]. We use a trainable ID embedding matrix $\mathcal{M}_u \in \mathcal{R}^{N_u \times Q}$ to represent each user ID as a low-dimensional vector, where N_u is the number of users and Q is the dimension of the ID embedding. For the user u, the one-hop ID embedding vector is denoted as $\mathbf{u}_e^O$.

3.3 Two-hop Graph Learning

The *two-hop graph learning* module mines the relatedness between neighbor users and news from the interaction graph. Additionally, for a given target user, neighbor users usually have different levels of similarity with her/his. The same situation exists between neighbor news. To utilize this kind of similarity, we aggregate neighbor news and user information with a graph attention network [22]. The utilized graph information here is heterogeneous, including both semantic representations and ID embeddings. In this *two-hop graph learning* module, there are also three parts: (1) Neighbor user ID representations; (2) Neighbor news ID representations; (3) Neighbor news semantic representations.

Neighbor User ID Representations. Since adding neighbor user information may complement target user representations, we aggregate the ID embeddings of neighbor users via an additive attention network. Given a user u and a list of D neighbor users $[u_{n_1}, u_{n_2}, ..., u_{n_D}]$, we first get their ID embeddings via the same user ID embedding matrix $\mathcal{M}_u$, which are denoted as $[\mathbf{m}_{u_1}, \mathbf{m}_{u_2}, ..., \mathbf{m}_{u_D}]$. Then the attention weight β_i^u of the i^{th} neighbor user is calculated as:

$$\beta_i^u = \frac{\exp(\mathbf{q}_u^T \tanh(\mathbf{U}_u \times \mathbf{m}_{u_i} + \mathbf{u}_u))}{\Sigma_{q=1}^D \exp(\mathbf{q}_u^T \tanh(\mathbf{U}_u \times \mathbf{m}_{u_q} + \mathbf{u}_u))}, \tag{4}$$

where $\mathbf{q}_u$, $\mathbf{U}_u$ and $\mathbf{u}_u$ are trainable parameters in the neighbor user attention network. The two-hop neighbor user ID representation $\mathbf{u}_e^T$ is then calculated as: $\mathbf{u}_e^T = \Sigma_{i=1}^D \beta_i^u \mathbf{m}_{u_i}$.

Neighbor News ID Representations. News clicked by the same user reveal certain preference of the user, thus may share some common characteristics. To model this kind of similarity, we utilize an attention network to learn neighbor news ID representations. For news n with a list of D neighbor news $[n_{n_1}, n_{n_2}, ..., n_{n_D}]$, we first transform neighbors with the news ID embedding matrix $\mathcal{M}_n \in \mathcal{R}^{N_n \times Q}$, where N_n is the number of news and Q is the dimension of the ID embedding. The output is $[\mathbf{m}_{n_1}, \mathbf{m}_{n_2}, ..., \mathbf{m}_{n_D}]$. Upon it, we apply an additive attention layer to combine neighbor ID embeddings into a unified output vector. The calculation of attention is similar with that in Eq.(4). The final two-hop neighbor news ID representation of news n is denoted as $\mathbf{n}_e^T$.

Neighbor News Semantic Representations. Although the ID embeddings of news are unique and inherently represent the neighbor news, they encode news information implicitly. Moreover, the IDs of some newly-sprung news may not be included in the predefined news ID embedding matrix $\mathcal{M}_n$. Thus, we propose to attentively learn their context representations via the transformer simultaneously. For the neighbor news list $[n_{n_1}, n_{n_2}, ..., n_{n_D}]$, the transformer outputs are $[\mathbf{v}_{n_1}, \mathbf{v}_{n_2}, ..., \mathbf{v}_{n_D}]$. Then the attention layer is applied to model varied importance of neighbor news. The final neighbor news semantic representation is the output of the attention layer, which is denoted as $\mathbf{n}_t^T$.

3.4 Recommendation and Model Training

The final representations of users and news are the concatenation of outputs from the *one-hop interaction learning* module and the *two-hop graph learning* module, i.e., $\mathbf{u} = [\mathbf{u}_t^O; \mathbf{u}_e^O; \mathbf{u}_e^T]$ and $\mathbf{n} = [\mathbf{n}_t^T; \mathbf{n}_e^T; \mathbf{n}_t^O]$. The rating score of a user-item pair is predicted by the inner product of user and item representation, i.e., $\hat{y} = \mathbf{u}^T \mathbf{n}$. Through this operation, the ID representations and semantic representations are optimized in the same vector space.

Motivated by [6, 36], we formulate the click prediction problem as a pseudo $\lambda + 1$ way classification task. We regard the clicked news as positive and the rest λ unclicked news as negative. We apply maximum likelihood method to minimize the log-likelihood on the positive class:

$$\mathcal{L} = -\sum_i \log\left(\frac{\exp(\hat{y}_i^+)}{\exp(\hat{y}_i^+) + \Sigma_{j=1}^{\lambda} \exp(\hat{y}_{i,j}^-)}\right), \tag{5}$$

where $\hat{y}_i^+$ is the predicted label of the i_{th} positive sample and $\hat{y}_{i,j}^-$ is the predicted label of the associated j_{th} negative sample.

4 EXPERIMENTS

4.1 Datasets and Experimental Settings

We constructed a large-scale real-world dataset by randomly sampling user logs from MSN News, [2] statistics of which are shown in Table 1. The logs were collected from Dec. 13rd, 2018 to Jan. 12nd, 2019 and split by time, with logs in the last week for testing, 10% of the rest for validation and others for training.

In our experiment, we construct D neighbors of the candidate news by random sampling from the clicked logs of its previous users. For the target user, since there exist massive neighbors users, we rank them according to the number of common clicked news

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

Table 1: Statistics of our dataset.

# users	242,175	# samples	32,563,990
# news	249,038	# positive samples	805,411
# sessions	377,953	# negative samples	31,758,579
# avg. words per title	10.99	# topics	285

with the target user. Then we pertain the top D users and use them as graph inputs. Here we set D to be 15 and use zero padding for cold-start user and newly-sprung news. [3] The dimensions of word embedding, topic embedding and ID embedding are set to 300, 128 and 128 respectively. We use the pretrained Glove embedding [18] to initialize the embedding matrix. There are 8 heads in the multi-head self-attention network, and the output dimension of each head is 16. The negative sampling ratio λ is set to 4. The maximum number of user clicked news is set to 50, and the maximum length of news title is set to 30. To mitigate overfitting, we apply dropout strategy [23] with the rate of 0.2 after outputs from the transformer and ID embedding layers. Adam [8] is set to be the optimizer and the batch size is set to be 128. These hyperparameters are selected according to the performances on the validation dataset.

For evaluation, we use the average AUC, MRR, nDCG@5 and nDCG@10 scores over all impressions. We independently repeat each experiment for 5 times and report the average results.

4.2 Performance Evaluation

In this section, we will evaluate the performance of our approach by comparing it with some baseline methods and a variant of our own method, which are listed as follow:

- *NGCF* [28]: a graph neural network based collaborative filtering method for general recommendation. They use ID embeddings as node representations.
- *LibFM* [21]: a feature based model for general recommendation using matrix factorization.
- *Wide&Deep* [3]: a general recommendation model which has both a linear wide channel and a deep dense-layer channel.
- *DFM* [11]: a neural news model utilizing an inception module to learn user features and a dense layer to merge them with item features.
- *DSSM* [6]: a sparse textual feature based model which learns news representation via multiple dense layers.
- *DAN* [38]: a CNN based news model which learns news representations from news titles. An attentional LSTM is used to learn user representations.
- *GRU* [17]: a deep news model using an auto-encoder to learn news representations and a GRU network to learn user representations.
- *DKN* [25]: a CNN based news model enhanced by the knowledge graph. They utilize news-level attention to form user representations.
- *GERL-Graph*: Our model without the *two-hop graph learning*.

For fair comparison, we extract the TF-IDF [7] feature from the concatenation of the clicked or candidate news titles and topics as sparse feature inputs for LibFM, Wide&Deep, DFM and DSSM. For DSSM, the negative sampling ratio is also set to 4. We try to tune all baselines to their best performances. The experimental results are summarized in Table 2, and we have several observations:

Table 2: The performance scores and standard variations of different methods. *The improvement is significant at the level p < 0.002.

Methods	AUC	MRR	nDCG@5	nDCG@10
NGCF [28]	55.45±0.16	17.19±0.05	17.23±0.10	22.08±0.09
LibFM [21]	61.83±0.10	19.31±0.06	20.45±0.08	25.69±0.08
Wide&Deep [3]	64.62±0.14	20.71±0.12	22.43±0.15	27.99±0.15
DFM [11]	64.72±0.19	20.75±0.14	22.60±0.20	28.22±0.19
DSSM [6]	65.49±0.18	20.93±0.13	22.93±0.22	28.65±0.27
DAN [38]	65.52±0.13	21.25±0.18	23.14±0.21	28.73±0.15
GRU [17]	65.69±0.19	21.29±0.10	23.16±0.11	28.75±0.11
DKN [25]	65.88±0.13	21.46±0.21	23.23±0.25	28.84±0.21
GERL-Graph	67.74±0.13	22.71±0.15	25.03±0.13	30.65±0.15
GERL	**68.55±0.12**	**23.33±0.10**	**25.82±0.14**	**31.44±0.12**

First, methods which represent news directly from news texts (e.g., DAN, GRU, DKN, GERL-Graph, GERL) usually outperform feature based methods (e.g., LibFM, Wide&Deep, DFM, DSSM). The possible reason is that although feature based methods learn news content, the useful information exploited from news texts is limited, which may lead to sub-optimal news recommendation results.

Second, compared with NGCF, which also exploits neighbor information in the graph, our method achieves better results. This is because NGCF is an ID-based collaborative filtering method, which may suffer from cold-start problem significantly. This result further proves the effectiveness of introducing textual understanding into graph neural networks for news recommendation.

Third, compared with other methods that involve textual content of news (e.g., DAN, GRU, DKN), our GERL-Graph can consistently outperform other baseline methods. This may because the multi-head attention in transformer module learns contextual dependency accurately. Moreover, our approach utilizes attention mechanism to select important words and news.

Fourth, our GERL approach which combines both textual understanding and graph relatedness learning outperforms all other methods. This is because GERL encodes neighbor user and news information by attentively exploiting the interaction graph. The result validates the effectiveness of our approach.

4.3 Effectiveness of Graph Learning

To validate the effectiveness of the *two-hop graph learning* module, we remove each component of representations in the module to examine its relative importance and illustrate the results in Figure 3. [4] Based on it, several observations can be made. First, adding the neighbor user information improves performances more significantly than adding neighbor news information. In our GERL-Graph approach, candidate news can be directly modeled through titles and topics, while target users are only represented by their clicked news. When the user history is sparse, they may not be well represented. Hence, adding IDs of neighbor users may assist our model to

[3] Due to limit of computational resources,we set D to be this moderate value.

[4] We use a trainable dense layer to transform vector $\mathbf{u}$ or $\mathbf{v}$ and keep the dimension uniform as before.

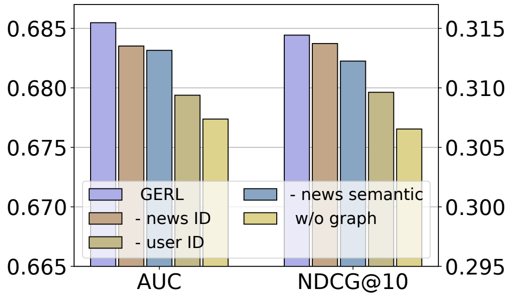


Figure 3: Effectiveness of two-hop graph learning.

learn better user representations. Second, the improvement brought by neighbor news semantic representations outweighs that brought by neighbour news ID. This is intuitive since titles of news contain more explicit and concrete meanings than IDs. Third, combining each part in the graph learning leads to the best model performance. By adding graph information both from neighbor users and news, our model forms better representations for recommendation.

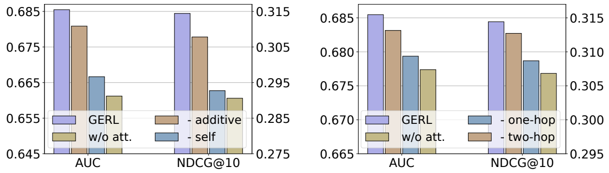
4.4 Ablation Study on Attention Mechanism

Next, we explore the effectiveness of two categories of attention by removing certain part of them. Instead, to keep dimensions of vectors unchanged, we use average pooling to aggregate information. First, we verify two types of attention inside the transformer in Figure 4(a). From it, we conclude that both the additive and the self attention are beneficial for news context understanding. This is because self-attention encodes interactions between words and additive attention helps to select important words. Among them, self-attention contributes more to improving model performances, as it models both short-distance and long-distance word dependency. Moreover, it forms diverse word representations with multiple attention heads. Also, we verify the model-level attention, e.g., attention inside the *one-hop interaction learning* and that in the *two-hop graph learning*. From Figure 4(b), we observe that the attention in the one-hop module is more important. One-hop attention selects important clicked news of users, thus helping model user preferences directly. Compared with that, two-hop attention models relative importance of neighbors, which may only represent interests implicitly. By using both attentions simultaneously, we obtain the best performances.

4.5 Hyperparameter Analysis

Here we explore the influences of two hyperparameters. One is the number of attention heads in the transformer module. Another one is the degree of graph nodes in the graph learning module.

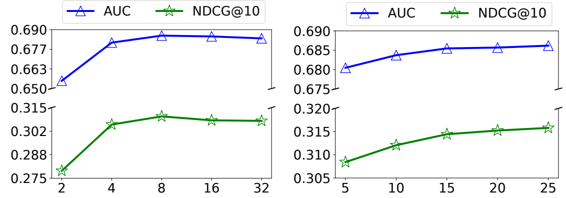
Number of Attention Heads. In the transformer module, the number of self-attention heads is crucial for learning context representations. We illustrate its influence in Figure 5(a). An evident increase can be observed when the number increases from 2 to 8, as the rich textual meanings may not be fully exploited when there are few heads. However, the performances drop a little when head number increases from 8. This may happen because news titles are concise and brief, thus too many parameters may be sub-optimal. Based on the above discussion, we set the number to be 8.



(a) Transformer attention.

(b) Model attention.

Figure 4: Effectiveness of attention mechanism.



(a) Attention head number.

(b) Node degree.

Figure 5: Influence of two hyperparameters.

Degree of graph nodes. In the graph learning module, the degree of user and item nodes decides how many similar neighbors our model will learn. We increase the node degree from 5 to 25 and showcase its influence in Figure 5(b). As illustrated, the performance improves when more neighbors are taken as model inputs, which is intuitive because more relatedness information from the graph is incorporated. Meanwhile, the increasing trend becomes smooth when the degree is larger than 15. Therefore, we choose a moderate value 15 as the number of node degree.

5 CONCLUSION

In this paper, we propose a graph enhanced representation learning architecture for news recommendation. Our approach consists of a *one-hop interaction learning* module and a *two-hop graph learning* module. The *one-hop interaction learning* module forms news representations via the transformer architecture. It also learns user representations by attentively aggregating their clicked news. The *two-hop graph learning* module enhances the representations of users and news by aggregating their neighbor embeddings via a graph attention network. Both IDs and textual contents of news are utilized to enrich the neighbor embeddings. Experiments are conducted on a real-world dataset, the improvement of recommendation performances validates the effectiveness of our approach.

ACKNOWLEDGMENTS

The authors would like to thank Microsoft News for providing technical support and data in the experiments. This work was supported by the National Key Research and Development Program of China under Grant number 2018YFC1604002, the National Natural Science Foundation of China under Grant numbers U1836204, U1705261, U1636113, U1536201, and U1536207, and the Tsinghua University Initiative Scientific Research Program under Grant number 20181080368.

REFERENCES

[1] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural News Recommendation with Long- and Short-term User Representations. In *ACL*. Association for Computational Linguistics, Florence, Italy, 336–345.

[2] Trapit Bansal, Mrinal Das, and Chiranjib Bhattacharyya. 2015. Content driven user profiling for comment-worthy recommendations of news and blog articles. In *RecSys*. ACM, 195–202.

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *DLRS*. ACM, 7–10.

[4] Guibing Guo, Jie Zhang, and Daniel Thalmann. 2014. Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowledge-Based Systems* 57 (2014), 57–68.

[5] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[6] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*. ACM, 2333–2338.

[7] Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* (2004).

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

[9] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. 1997. GroupLens: applying collaborative filtering to Usenet news. *Commun. ACM* 40, 3 (1997), 77–87.

[10] Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. 2011. SCENE: a scalable two-stage personalized news recommendation system. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 125–134.

[11] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2018. Towards Better Representation Learning for Personalized News Recommendation: a Multi-Channel Deep Fusion Approach.. In *IJCAI*. 3805–3811.

[12] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. 2014. Facing the cold start problem in recommender systems. *Expert Systems with Applications* 41, 4 (2014), 2065–2073.

[13] Guang Ling, Michael R Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 105–112.

[14] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *IUI*. ACM, 31–40.

[15] Yuanhua Lv, Taesup Moon, Pranam Kolari, Zhaohui Zheng, Xuanhui Wang, and Yi Chang. 2011. Learning to model relatedness for news recommendation. In *WWW*. ACM, 57–66.

[16] Benjamin Marlin and Richard S Zemel. 2004. The multiple multiplicative factor model for collaborative filtering. In *ICML*. ACM, 73.

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

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

[19] Owen Phelan, Kevin McCarthy, Mike Bennett, and Barry Smyth. 2011. Terms of a feather: Content-based news recommendation and discovery using twitter. In *ECIR*. Springer, 448–459.

[20] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. 2017. Social collaborative viewpoint regression with explainable recommendations. In *WSDM*. ACM, 485–494.

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

[22] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *WSDM*. ACM, 555–563.

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR* 15, 1 (2014), 1929–1958.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.

[25] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *WWW*. International World Wide Web Conferences Steering Committee, 1835–1844.

[26] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In *25th ACM SIGKDD (KDD '19)*. ACM, New York, NY, USA, 968–977.

[27] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *KDD*. 950–958.

[28] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR 2019, Paris, France, July 21-25, 2019*. 165–174.

[29] Xuejian Wang, Lantao Yu, Kan Ren, Guanyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In *KDD*. ACM, 2051–2059.

[30] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with attentive multi-view learning. *arXiv preprint arXiv:1907.05576* (2019).

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

[32] Chuhan Wu, Fangzhao Wu, Mingxiao An, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Topic-Aware News Representation. In *ACL*. 1154–1159.

[33] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.

[34] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. 2019. Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation. In *SIGIR*.

[35] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*. ACM, 974–983.

[36] Shuangfei Zhai, Keng-hao Chang, Ruofei Zhang, and Zhongfei Mark Zhang. 2016. Deepintent: Learning attentions for online advertising with recurrent neural networks. In *KDD*. ACM, 1295–1304.

[37] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *WWW*. 167–176.

[38] Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. DAN: Deep Attention Neural Network for News Recommendation. In *AAAI*, Vol. 33. 5973–5980.