WG4Rec: Modeling Textual Content with Word Graph for News Recommendation

Shaoyun Shi[1], Weizhi Ma[2], Zhen Wang[1], Min Zhang[1]*, Kun Fang[3], Jingfang Xu[3],
Yiqun Liu[1], and Shaoping Ma[1]
[1]Department of Computer Science and Technology, Institute for Artificial Intelligence,
Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
[2]Institute for AI Industry Research (AIR), Tsinghua University, Beijing 100084, China
[3]Sogou Inc., Beijing 100084, China
shisy17@mails.tsinghua.edu.cn,z-m@tsinghua.edu.cn

ABSTRACT

News recommendation plays an indispensable role in acquiring daily news for users. Previous studies make great efforts to model high-order feature interactions between users and items, where various neural models are applied (e.g., RNN, GNN). However, we find that seldom efforts are made to get better representations for news. Most previous methods simply adopt pre-trained word embeddings to represent news and also suffer from cold-start users.

In this work, we propose a new textual content representation method by building a word graph for recommendation, which is named WG4Rec. Three types of word associations are adopted in WG4Rec for content representation and user preference modeling, namely: 1) *semantically-similar* according to pre-trained word vectors, 2) *co-occurrence* in documents, and 3) *co-click* by users across documents. As extra information can be unified by adding nodes/edges to the word graph easily, WG4Rec is flexible to make use of cross-platform and cross-domain context for recommendation to alleviate the cold-start issue. To the best of our knowledge, it is the first attempt that using these relationships for news recommendation to better model textual content and adopt cross-platform information. Experimental results on two large-scale real-world datasets show that WG4Rec significantly outperforms state-of-the-art algorithms, especially for cold users in the online environment. Besides, WG4Rec achieves better performances when cross-platform information is utilized.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

word graph; news recommendation; neural networks

ACM Reference Format:
Shaoyun Shi, Weizhi Ma, Zhen Wang, Min Zhang, Kun Fang, Jingfang Xu, Yiqun Liu, and Shaoping Ma. 2021. WG4Rec: Modeling Textual Content with Word Graph for News Recommendation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482401

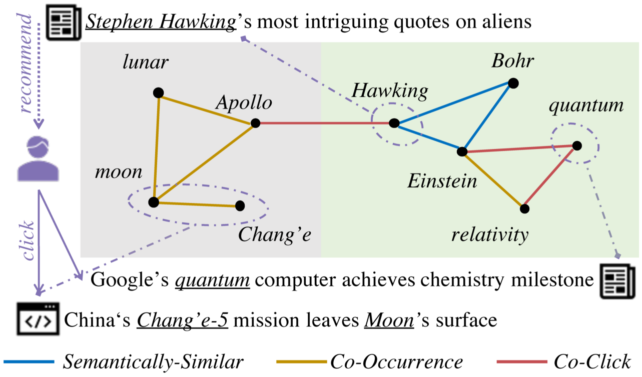

Stephen Hawking's most intriguing quotes on aliens
recommend
lunar
Apollo
Hawking
Bohr
quantum
moon
Einstein
Chang'e
relativity
click
Google's quantum computer achieves chemistry milestone
China's Chang'e-5 mission leaves Moon's surface
Semantically-Similar
Co-Occurrence
Co-Click

Figure 1: An example of news recommendation based on the word graph, which contains three types of global word relations: *semantically-similar*, *co-occurrence*, and *co-click*.

1 INTRODUCTION

Recommender systems are widely adopted in online news platforms, e.g., Google News, Yahoo News. Unlike other scenarios like e-commerce, in which collaborative filtering (CF) usually achieves excellent performance with sufficient interaction history, the news recommendation scenario has lots of cold items that constrain the performances of CF-based algorithms. Therefore, content information, especially textual content, is the key to modeling both users and news in this scenario.

Previous studies have paid much attention to modeling interactions between users and items, such as introducing personalized attention or recurrent neural networks (RNN) over interaction sequences [45, 47, 54] or applying graph neural networks (GNN) on a user-item bipartite graph [8, 15, 16]. However, there are still two challenges: 1) Lack of powerful content embedding. Previous methods model news documents based on pre-trained word embeddings. Although these methods have achieved remarkable results, we argue that the pre-trained word embeddings are insufficient to

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8446-9/21/11...$15.00
https://doi.org/10.1145/3459637.3482401

* Corresponding Author

This work is supported by the National Key Research and Development Program of China (2018YFC0831900), Natural Science Foundation of China (Grant No. 62002191, 61672311, 61532011) and Tsinghua University Guoqiang Research Institute. This research is partly supported by the Tsinghua-Sogou Tiangong Institute for Intelligent Computing.

model various types of relations between words and associations among news content, which may constrain the downstream recommendation performance. 2) User cold-start problem. Users with less history (cold-start users) cause the well-known user cold-start problem in real systems, which is more serious for methods that simply adopt news content embedding methods. In such a scenario, better utilization of limited content data and users' interacted content from other domains helps model users' interests. Although previous work [46] noticed that cross-platform content is helpful, we argue that current methods can still be improved, especially by explicitly exploring relations between texts.

To tackle the above challenges, we propose to construct a word graph to better model textual content, which utilizes *semantically-similar*, *co-occurrence*, and *co-click* relationships between words for news recommendation. An example of the three types of relationships is shown in Figure 1, which is a subgraph of our constructed word graph on real-world datasets. 1) *Semantically-similar* words provide users with additional information about similar content in terms of semantics. "*Hawking*", "*Einstein*", and "*Bohr*" have close vector representations because they have similar semantic meanings (famous scientists). Users who are interested in "*Hawking's*" talk may also read the biography about "*Einstein*". 2) But some related words can be semantically different. "*Relativity*" is a great theory proposed by "*Einstein*". They frequently occur together but are not close in the semantic space. If the recommendation model realizes that they are *co-occurrence* words, it will be easier to capture that one likes books about "*relativity*" may also click news about "*Einstein's*" birth anniversary. 3) Except for *semantically-similar* and *co-occurrence* words, which can be directly extracted from textual corpus, there are also *co-click* signals to provide word-level CF. This non-semantic relationship helps explore users' interests according to others' history. *Co-click* means users often interact with both of the two words in their history even if they are in different documents, which may corresponds to different domains. For example, "*Hawking*" and "*Apollo*" are co-clicked by a large number of users. The reason may be that "*Hawking*" puts forward many views about space and aliens, and users who read such quotes are also interested in space missions. These word relations naturally link documents in various domains, and thus enhance textual content modeling. For example, as shown in Figure 1, we can explore the potential relationship between webpages about "*Chang'e*" and news documents about "*Hawking*" according to the word graph.

Our work aims to build a word graph to explicitly capture various types of word relations and associations among textual content for news recommendation, which is hardly captured by only utilizing pre-trained word vectors. Besides, we propose a novel news recommendation framework, named WG4Rec, to model textual content with the proposed word graph. Firstly, we construct a word graph according to the textual content and user interactions with the fore-mentioned three types of connections. The similarity of word vectors and *co-occurrence* of words help interpret words' semantic meanings and thus find related content. *Co-click* signals conduct CF on word-level and explore users' potential interest according to others' interactions. Secondly, WG4Rec uses RNN and attention mechanisms to model the user history and dynamically adjust the information resource. Note that the word graph is naturally capable of bridging users' interacted textual content even from multiple sources and cross-domains, and in turn, cross-scenario documents enrich the word graph construction and better reveal users' preferences, especially for cold-start users. Two large-scale datasets collected from real-world news recommender systems are used to verify the effectiveness of WG4Rec. Results show that WG4Rec significantly outperforms state-of-the-art algorithms, especially for cold users in the online environment. In one of the datasets, WG4Rec is applied to both user interacted news and clicked webpages in SERP and achieves more encouraging improvements.

The main contributions are summarized as follows:

(1) We propose to construct a word graph to explicitly model word relationships for better content modeling in news recommendation, including *semantically-similar*, *co-occurrence*, and *co-click* of words. The construction is free from external knowledge data and is flexible to combine cross-platform textual information.

(2) A novel news recommendation framework named WG4Rec is designed. The proposed word graph contributes significantly to modeling textual content from multiple resources and user preference modeling, especially for cold-start users.

(3) Experimental results on two real-world datasets and an online platform show that WG4Rec outperforms some state-of-the-art algorithms on both warm and cold users.

2 RELATED WORK

2.1 Neural News Recommendation

Deep learning has shown remarkable results in many research and application areas, and researchers propose some neural news recommendation models in recent years. RA-DSSM improves DSSM [17] by applying attention mechanism and RNN, and achieves better performance than previous algorithms [25]. Okura et al. propose an embedding-based news recommendation model for large-scale users based on LSTM [14] which achieves excellent performance on Yahoo logs [29]. Weave&Rec models users' historical interactions by a 3D-CNN [21], and DKN improves news content modeling by introducing some entity embeddings on knowledge graph [41]. To help the model attend to important words and news articles, NPA applies both word- and news-level personalized attention mechanisms [45], and NRMS uses multi-head self-attention networks to model the interactions between words and capture the relatedness between the news [47]. Similarly, DAN uses attention-based parallel CNN for aggregating users' interest features and attention-based RNN to capture richer hidden sequential features of users' clicks, and it combines these features for news recommendation [54]. Recently there is also work considering user-news interactions as a graph. In GERL, users and news are both viewed as nodes in a bipartite graph constructed from historical user click behaviors [8]. GNewsRec constructs a heterogeneous user-news-topic graph to model user-item interactions [15]. It alleviates the sparsity of user-item interactions and models both long-term and short-term user interest. GNUD improves users and news representations by fully considering the high-order connectivities and latent preference factors with unsupervised preference disentanglement [16].

However, these models simply use pre-trained word embeddings and only focus on modeling high-order feature interactions between users and items, where relationships among words are modeled implicitly or ignored and cold-start users are not well handled.

2.2 Cold-Start Recommendation

Previous models focus on users' interacted news, while new users with no history are coming to the system every day, which leads to the cold-start problem. One of the typical ways to handle the cold-start problem is by introducing external information. For example, some approaches utilize users' profiles and combine the CF and content-based recommendation [35, 39, 49]. Social networks are another valuable information because users may also like items their friends interact with [1, 18, 53]. Cross-domain information, such as user behaviors on other platforms, also helps [6]. Ma et al. [27] introduce users' content information from other platforms (user posts on Twitter) to learn extra user features for the recommendation. In news recommendation literature, researchers focus more on the news content, and few have considered other types of textual content. Although NRHUB aggregates heterogeneous user behaviors by attention networks [46], it ignores the word-level interactions.

Although it is challenging to link textual content in various scenarios, we believe modeling relationships among words is a possible solution as words are shared and their relations are stable. While it has not been studied by previous news recommendation methods.

2.3 Graph of Words

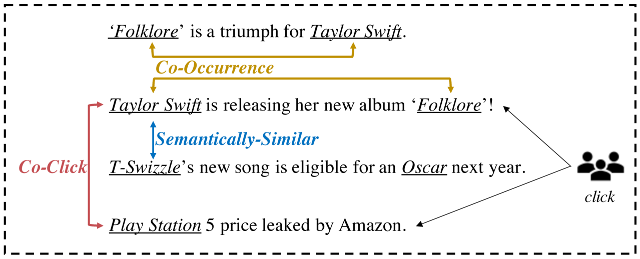
One of the closest ideas to model word relations is WordNet, a lexical database for English [7]. It is constructed by lots of experts in linguistics. Although WordNet links over 150,000 words into semantic relations, it is not designed specifically for the recommendation, and a large number of words in news documents are not covered, especially neologisms. Besides, some researchers build graphs of words for speech recognition [30, 44] or machine translation [5, 38]. Such a word graph is also called a word lattice. While it is designed mainly for the language model and has limited applications.

In recent years, knowledge graphs have attracted much attention in both industry [36] and research [42] areas. Some recent recommendation models also try to utilize knowledge graphs [40, 43]. The knowledge graph represents a collection of interlinked descriptions of entities – objects, events, or concepts. RippleNet stimulates the propagation of user preferences over the set of knowledge entities by automatically and iteratively extending a user's potential interests along with links in the knowledge graph [40]. However, knowledge graphs, such as ConceptNet [37], focus on entity words, and the construction usually requires plenty of human effort and expert knowledge [31], which may limit their applications. In contrast, we aim to construct a word graph for recommendation with some intuitive but effective strategies, and it covers most words in the corpus.

3 WG4REC MODEL

In this work, we propose a novel news recommendation framework based on a word graph (WG4Rec) to tackle the challenges mentioned above. The framework of WG4Rec is shown in Figure 3, which mainly has three modules:

(1) Word Graph Construction. The word graph is constructed specifically for the recommendation based on content information and user interactions, and relations are extracted offline before the model training.


'Folklore' is a triumph for Taylor Swift.
Co-Occurrence
Taylor Swift is releasing her new album 'Folklore'!
Semantically-Similar
Co-Click
T-Swizzle's new song is eligible for an Oscar next year.
click
Play Station 5 price leaked by Amazon.

Target Word: Taylor Swift

Word Relations	*Candidates*	*Filtering*	*Relational Words*
Semantically-Similar:	*[all words in corpus]*	*Word2Vec similarity*	T-Swizzle, ...
Co-Occurrence:	triumph, release, album, Folklore, ...	*tf-idf*	Folklore, album, ...
Co-Click:	Play Station, price, leak, Amazon, ...	*tf-idf*	Play Station, ...

Figure 2: An illustration of the word graph construction. Relational words of target word "*Taylor Swift*" are extracted according to the text corpus and user-item interactions.

(2) Word Graph Modeling. A graph neural network is adopted to explicitly model various relationships among words and works as a fundamental component to provide enhanced word vectors for further content representation and user preference modeling.

(3) User Preference Profiling. A multi-level structure first uses a word attention network to generate document embeddings. Then, attentional RNNs over interaction sequences are utilized to model the interacted document sequences. A preference attention network combines user preferences from different scenarios at last.

The entire framework is trained end-to-end under the supervision of the recommendation target.

3.1 Word Graph Construction

Although various textual content types, like news and webpages, have different lengths or styles, they usually share the same set of words. Unlike knowledge graphs, in which external knowledge and human effort is necessary, WG4Rec constructs the word graph according to textual content and user interactions. It improves textual content modeling specifically for the recommendation. An illustration of the word graph construction is shown in Figure 2. For a given target word and relation, we first generate a collection of candidate words according to the text corpus and user-item interactions, and then apply some sorting and filtering strategies to obtain the top-related words. In this work, the word graph includes three typical types of relationships: *semantically-similar*, *co-occurrence*, and *co-click*.

• *Semantically-Similar*. Using pre-trained word vectors is a popular way of word representation in various deep models. Words with close semantic meanings usually have similar vectors. So words with high Word2Vec similarity should be *semantically-similar*. For a central word w_i, suppose the word with the largest Word2Vec similarity is $w_{i,1}^{wv}$, i.e.,

$$w_{i,1}^{wv} = max_{w_j}\{\frac{\mathbf{w}_i^\top \mathbf{w}_j}{\|\mathbf{w}_i\|\|\mathbf{w}_j\|}\} \quad (1)$$

where $\mathbf{w}_i$ is the vector of word w_i. E.g., *"Taylor Swift"* and *"T-Swizzle"* (the nickname of *"Taylor"*) are *semantically-similar* in Figure 2. To control connections of each central word w_i, top n nearest neighbor words are considered, i.e., there are n edges from $w_{i,1}^{wv}, w_{i,2}^{wv}, ..., w_{i,n}^{wv}$ to w_i in the word graph with the relation type *semantically-similar*. The edges are directional because w_i may be not in the top-n nearest neighbours of $w_{i,j}^{wv}$.

- *Co-Occurrence*. Unlike *semantically-similar* words, two words in *co-occurrence* relation can have totally different kinds of concepts, such as a singer's name and his/her albums (*"Taylor Swift"* and *"Folklore"* in Figure 2). In this work, we calculate *co-occurrence* words for a central word w_i by the following steps. Firstly, for each word, we count the frequency of other words occurring around the central word (within a certain range) in the same document. Let $o_{i,j}$ denote the frequency of w_j occurs together with w_i in all data ($o_{i,j} = o_{j,i}$). Secondly, each word w_i is represented as a vector $\mathbf{o}_i = [o_{i,1}, o_{i,2}, ..., o_{i,|W|}]^\top$, where W is the set of all distinct words. $\{w_j | o_{i,j} > 0\}$ is the set of candidate *co-occurrence* words. Regarding each $\mathbf{o}_i$ as a virtual "document", there are $|W|$ "documents" in total, each of which consists of candidate words of corresponding central word w_i. Thirdly, tf-idf is calculated to evaluate the importance of candidates. Formally,

$$tf_{i,j} = \frac{o_{i,j}}{\sum_{k=1}^{|W|} o_{i,k}}, \quad idf_j = lg\frac{|W|}{|\{o_{k,j} | \forall k, o_{k,j} > 0\}|} \tag{2}$$
$$tf\text{-}idf_{i,j} = tf_{i,j} \times idf_j$$

A high $tf\text{-}idf_{i,j}$ means that w_j frequently occurs with w_i, and w_j does not *co-occurrence* with other words so often. Finally, for each central word w_i, at most n words with the largest tf-idf scores are regarded as neighbor words with *co-occurrence* relation in the graph. It is denoted by n edges from $w_{i,1}^{oc}, w_{i,2}^{oc}, ..., w_{i,n}^{oc}$ to w_i where

$$w_{i,1}^{oc} = max_{w_j}\{tf\text{-}idf_{i,j}\}, \tag{3}$$

and they are also directional.

- *Co-Click*. Item *co-click* information is usually regarded as an efficient and effective way to provide accurate recommendations. But in news recommendation, most of the news documents are cold items, where CF methods do not work. So we propose to utilize user interactions with news and textual content from other scenarios to extract *co-click* relations of words, which is generally stable even in cross-domains and works for cold items. *Co-click* here means users often click both of the two words in their history even if they are in different documents. In detail, for each pair of words, w_i and w_j, we count the pair's *co-click* frequency $c_{i,j} = c_{j,i}$ that the two words occur in two different documents but clicked by the same user. Then similar as *co-occurrence*, by replacing $o_{i,j}$ with $c_{i,j}$ in Equation 2, at most top n related *co-click* words of a central word w_i are obtained, denoted by $w_{i,1}^{cl}, w_{i,2}^{cl}, ..., w_{i,n}^{cl}$. It means that if a user clicks content about word w_i, he/she may also be interested in content about $w_{i,j}^{cl}$, according to the interactions in the data. Like the *"Taylor Swift"* and *"Play Station"* in Figure 2, *co-click* of words can model relations that may not be captured by semantic meanings. It discovers the potential interests of users and alleviates the information cocoons problem.

In summary, three types of relations are constructed for each word to obtain messages passing from their neighbors in the graph.

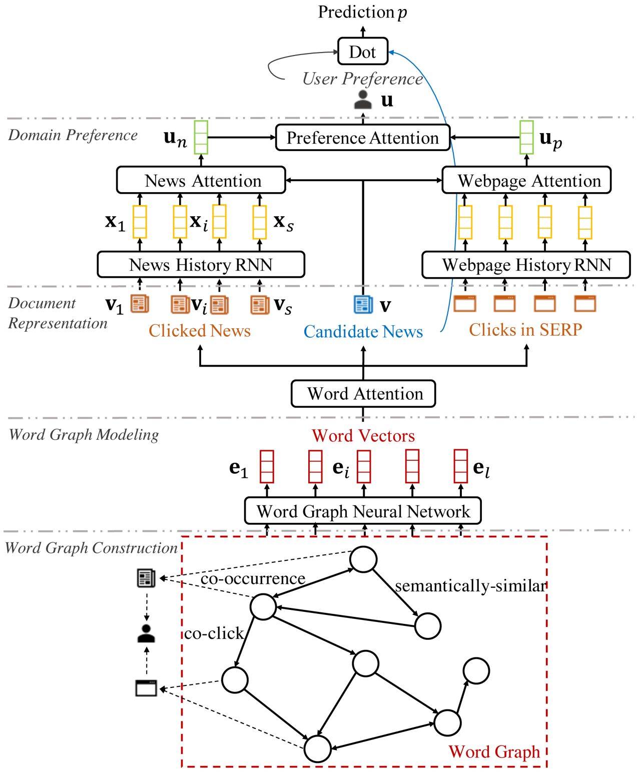


Figure 3: An illustration of the WG4Rec framework. Each word in the graph considers its top related neighbors in different relations, so the graph is directional.

Besides, the construction method is not limited to news documents. Various textual content in cross-platforms, such as users' clicked webpages, can also be utilized together to construct the word graph and enrich these types of word relations.

3.2 Word Graph Modeling

Words and their relations are used to retrieve related content for recommendation and link content from various scenarios. It is vital to generate good word representations according to the word graph. Graph neural networks (GNN), which has been verified powerful to model graph data [48, 52], is used in WG4Rec to model relationships among words. The word graph may contain hundreds of thousands of nodes (words) and millions of edges, so WG4Rec takes the GraphSAGE algorithm to learn a function that generates node embeddings by aggregating sampled neighbors of a central node [11]:

$$\mathbf{h}_i^t = \sigma\Big(\mathbf{W}_g\big(\mathbf{h}_i^{t-1} \oplus \text{AGGREGATE}(\{\mathbf{h}_j^{t-1}, \forall w_j \in \mathcal{N}_{w_i}^*\})\big)\Big) \tag{4}$$

where $\mathcal{N}_{w_i}$ is the neighbor set of the central word w_i, and $w_j \in \mathcal{N}_{w_i}$ means there is an edge from w_j to w_i. $\mathcal{N}_{w_i}^*$ is a sampled set of $\mathcal{N}_{w_i}$ with size k, which is re-sampled every training step. $\mathbf{h}_i^t \in \mathbb{R}^d$ is the representation of word w_i in the t-th layer of GNN, which comes from its representation $\mathbf{h}_i^{t-1}$ and neighbors $\mathbf{h}_j^{t-1}$ in previous layer $t-1$, and initially $\mathbf{h}_i^0 = \mathbf{w}_i$. $\mathbf{W}_g \in \mathbb{R}^{d \times 2d}$ is used to aggregate the neighbor representation with the central word. σ is the non-linear activation function. AGGREGATE$(\cdot)$ is the aggregating function of neighbors, which is implemented by:

$$
\begin{aligned}
\mathbf{q}_g^t &= \sigma\left(\sum_{w_j \in \mathcal{N}^*_{w_i}} \mathbf{Q}_g \mathbf{h}_j^{t-1}\right), \quad \mathbf{k}_j^t = \sigma\left(\mathbf{K}_g \mathbf{h}_j^{t-1}\right) \\
a_j^t &= \frac{exp(\mathbf{q}_g^{t\top} \mathbf{k}_j^t)}{\sum_{w_k \in \mathcal{N}^*_{w_i}} exp(\mathbf{q}_g^{t\top} \mathbf{k}_k^t)} \\
\mathbf{h}^t_{\mathcal{N}^*_{w_i}} &= \sum_{w_j \in \mathcal{N}^*_{w_i}} a_j^t \mathbf{h}_j^{t-1}
\end{aligned}
\tag{5}
$$

where $\mathbf{Q}_g, \mathbf{K}_g \in \mathbb{R}^{d\times d}$ are parameters of the aggregation function. $\mathbf{h}^t_{\mathcal{N}^*_{w_i}}$ is the representation of aggregated neighborhood of w_i. The attention mechanism here evaluates the importance of neighbor nodes and their relatedness to the central word. Thus w_i's final enriched vector after T layers of GNN is

$$
\mathbf{e}_i = \mathbf{h}_i^T = \sigma\Big(\mathbf{W}_g\big(\mathbf{h}_i^{T-1} \oplus \mathbf{h}^T_{\mathcal{N}^*_{w_i}}\big)\Big) \tag{6}
$$

The enhanced word vectors are further used to represent various textual content. An end-to-end training updates parameters of GNN under the supervision of the recommendation target. Word vectors $\mathbf{w}_i$ are also updated during training.

Note that there are also other ways to model the word graph, such as Graph Convolutional Neural Network (GCN) [24] and its variants considering directed edges [20] or relation types [34, 51]. However, this kind of nodes aggregation takes all neighbors and costs too much computation resources, which is not suitable for large-scale graphs. WG4Rec applies the GNN layer described in Equation 4 and 5 for better efficiency, and its performance is verified to be good. Another advantage is that GraphSAGE can quickly generate embeddings when edges or nodes are added/deleted by the learned aggregation function, so it is convenient for updating the word embeddings without retraining the model.

3.3 User Preference Profiling

The word graph works as a fundamental component, which naturally links various textual content. For example, in one of our datasets, in addition to users' clicked news, WG4Rec also utilizes their clicked webpages in SERP. Interactions in both scenarios show users' preferences but from different aspects. As a result, WG4Rec uses two branches with similar network structures to model user preferences on the two scenarios, and finally combines them. Users' preferences from other domains can be unified in a similar way.

3.3.1 Document Representation. To efficiently model the interactions among words and automatically retrieve valuable information for document representation, a word-level self-attention is applied to generate the representations of documents (news and webpages here). Formally, suppose a document v is represented by a sequence of words $w_1, w_2, ..., w_l$, and their enriched word representations are $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_l$, where l is the text length. Then the news representation $\mathbf{v}$ is obtained by:

$$
\begin{aligned}
\mathbf{q}_w &= \sigma\left(\sum_{i=1}^{l} \mathbf{Q}_w \mathbf{e}_i\right), \quad \mathbf{k}_{e_i} = \sigma\left(\mathbf{K}_w \mathbf{e}_i\right) \\
a_{e_i} &= \frac{exp(\mathbf{q}_w^\top \mathbf{k}_{e_i})}{\sum_{j=1}^{l} exp(\mathbf{q}_w^\top \mathbf{k}_{e_j})} \\
\mathbf{v} &= \sigma\left(\mathbf{W}_w \sum_{i=1}^{l} a_{e_i} \mathbf{e}_i\right)
\end{aligned}
\tag{7}
$$

where $\mathbf{Q}_w, \mathbf{K}_w, \mathbf{W}_w \in \mathbb{R}^{d\times d}$ are the parameters of word-level self-attention. Word vectors $\mathbf{e}_i$ are the outputs of GNN, which are shared among different scenarios. Thus word-level attention parameters are also shared to model documents composed by the same group of word representations. $\mathbf{v}$ is the textual content representation of the document. In real applications, it can further be unified with other features or pre-trained embeddings.

3.3.2 Domain Preference. Users' interaction patterns on different platforms or scenarios vary from each other. To model news and webpages interaction sequences, WG4Rec uses the same network structure but different parameters. Taking the news part (left branch in Figure 3) as an example, suppose a user u has interacted with s news $v_1, v_2, ..., v_s$ just before the recommendation request, then the user preference in news platform $\mathbf{u}_n$ is modeled by a GRU and a self-attention:

$$
\begin{aligned}
\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_s\} &= \mathrm{GRU}(\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_s\}) \\
\mathbf{q}_v &= \sigma(\mathbf{Q}_x \mathbf{v}), \quad \mathbf{k}_{x_i} = \sigma(\mathbf{K}_x \mathbf{x}_i) \\
a_{x_i} &= \frac{\exp(\mathbf{q}_v^\top \mathbf{k}_{x_i})}{\sum_{j=1}^{s} \exp(\mathbf{q}_v^\top \mathbf{k}_{x_j})} \\
\mathbf{u}_n &= \sum_{i=1}^{s} a_{x_i} \mathbf{x}_i
\end{aligned}
\tag{8}
$$

where $\mathbf{Q}_x, \mathbf{K}_x, \mathbf{W}_x \in \mathbb{R}^{d\times d}$ are the parameters of document-level attention in the news scenario. GRU is the Gated Recurrent Unit [3] to model the news interaction sequences. Some previous sequential recommendation work has verified that considering sequential information is helpful for recommendation performance [13, 19, 26]. GRU encodes sequential information and gives a representation $\mathbf{x}_i$ at each timestamp. Then document-level attention is used to form the user preference dynamically. Note that the candidate item vector, i.e., $\mathbf{v}$, is used to calculate the attention query vector $\mathbf{q}_v$. This design makes the attention network retrieve related information from users' interaction history for the current prediction target and automatically adjust documents weights. Finally, the user preference in news platforms $\mathbf{u}_n$ is generated. His/her interactions with other types of textual content are modeled by the same network structure as Equation 8 but with different parameters. The preference representation from clicks in SERP is denoted as $\mathbf{u}_p$.

3.3.3 User Preference. Users have different numbers of interactions with various textual content. Some are heavy news recommendation users but seldom search and click webpages in the system, and others may be the opposite. It is essential to evaluate the richness and usefulness of information from different scenarios, so a preference attention network is used to integrate users' preferences from multiple scenarios dynamically. Suppose $\{\mathbf{u}_{d_i}\}$ is the set of user preferences from various scenarios d_i. For example, it can be $\{\mathbf{u}_n, \mathbf{u}_p\}$ from interacted news and clicks in SERP. Formally, the user preference vector $\mathbf{u}$ is calculated by:

$$\begin{aligned} c_{d_i} &= \mathbf{d}^\top \sigma(\mathbf{W}_d \mathbf{u}_{d_i} + \mathbf{b}_d), \\ a_{d_i} &= \frac{exp(c_{d_i})}{\sum_{d_j} exp(c_{d_j})}, \\ \mathbf{u} &= \sum_{d_i} a_{d_i} \mathbf{u}_{d_i} \end{aligned} \tag{9}$$

where $\mathbf{d} \in \mathbb{R}^d, \mathbf{W}_d \in \mathbb{R}^{d \times d}, \mathbf{b}_d \in \mathbb{R}^d$ are parameters of the preference attention. Although we focus on the cold-start scenario in this work, in order to model long-term interests for warm users in real applications, users' id embeddings can also be included in $\{\mathbf{u}_{d_i}\}$.

3.4 Model Training

The dot product of user preference vector $\mathbf{u}$ and candidate item vector $\mathbf{v}$ is the final prediction score of WG4Rec, i.e.:

$$p = \mathbf{u}^\top \mathbf{v} \tag{10}$$

For top-n recommendation task, WG4Rec uses the pair-wise training strategy [33]. For each positive interaction v^+, we randomly sample an item the user dislikes or has never interacted with as the negative sample v^- in each epoch. Then, the loss function is:

$$L_{bpr} = -\sum_{v^+} \log\left(sigmoid(p_{v^+} - p_{v^-})\right) + \lambda_\Theta \|\Theta\|_F^2 \tag{11}$$

where p_{v^+} and p_{v^-} are the prediction results of v^+ and v^-, respectively, and $\lambda_\Theta \|\Theta\|_F^2$ is the ℓ_2-regularization. The loss function encourages predictions of positive interactions to be higher than the negative samples.

For click-prediction task, we apply the binary cross-entropy loss for each training sample:

$$L_{cr} = -\sum_{v^+} \log\left(sigmoid(p_{v^+})\right) - \sum_{v^-} \log\left(1 - sigmoid(p_{v^-})\right) + \lambda_\Theta \|\Theta\|_F^2 \tag{12}$$

3.5 Discussion

WG4Rec separates the word graph construction and model learning into two stages and uses different branches to model various document preferences. This design has the following advantages:

- The word graph construction is entirely offline. No training is needed in the construction process given the pre-trained word vectors. The graph can be modified and updated without retraining the downstream recommendation model, which is valuable and efficient for real-world systems.
- Although we use the Word Attention to form the document representation, other more powerful text representation methods such as Transformers or BERT [4] can be easily integrated with the word graph by taking word vectors as inputs.
- Although the illustration of WG4Rec is based on interacted news and clicks in SERP, it is not limited to these two scenarios. Interacted documents in other platforms or scenarios can be easily incorporated.
- The framework is flexible to utilize external knowledge by introducing more nodes/edges and edge types into the word graph.

4 EXPERIMENTAL SETTINGS

WG4Rec is evaluated on both top-n recommendation and click prediction tasks. We run the experiments with 5 different random seeds and report the average results and standard errors. Codes and datasets can be found in https://github.com/THUIR/WG4Rec.

4.1 Top-N Recommendation

The top-n recommendation dataset is collected from a real-world system by Sogou [1], including users' interactions on news recommendation service and their clicked webpages in the search engine result pages (SERP). The news clicks and web search logs are collected and limited in the same service provider (through a mobile application). The logs do not contain clicks outside SERP, such as clicks inside browsed pages. And there is an incognito mode in which behaviors are not tracked. Moreover, no users' personal profiles are recorded here. Users will choose whether to allow the app to track their behaviors in the privacy settings (lying at the first of preference settings). All data collection has users' permissions and meets the requirements of relevant laws and regulations. In this work, news and webpages are represented by title words because users mainly decide whether to click based on the titles, which are high-quality summaries of the content. Some detailed information about the dataset and constructed word graph is shown in Table 1.

Table 1: Statistics of the top-n recommendation dataset and word graph (WG). $\bar{l}$ denotes the average title length.

# users	68,896	# news impressions	8,743,352
# news	588,912	# news clicks	1,308,487
# webpages	481,318	# webpage clicks	768,009
news $\bar{l}$	8.1	# WG nodes	500,000
webpage $\bar{l}$	10.13	# WG edges	45,779,395

The dataset includes nine days of logs, and we use the first eight days for training and randomly split the last day for validation and test. In addition to the known negative items in the impression list, we sample some of users' non-interacted items during the validation and testing so that the ratio of positive and negative items is 1 : 99.

WG4Rec is compared with the following baselines.

- Id-based methods, without modeling of content: **BPRMF** (2009) is one of the most famous traditional matrix factorization models [33]; **GRU4Rec** (2015) is a deep sequential recommendation method which uses a GRU to model users' interaction sequences [13].
- Content-based methods, which take users' clicked and candidate news words as features in our experiments: **Wide&Deep** (2016) combines the deep neural network and linear models, and takes both content features and id embeddings as inputs [2]; **NFM** (2017) is a neural factorization machine which uses a bi-interaction layer to model feature interactions [12].
- Hybrid methods, which combines CF and CB: **ACCM** (2018) works on both warm and cold scenarios, and the model adopts a "Cold-Sampling" (CS) strategy to help the attention network learn how to handle the cold data [35].
- News recommendation methods, recently proposed specifically for news recommendation: **NPA** (2019) is a powerful news recommendation models, but it does not consider users' interactions from other scenarios [45]; **NRHUB** (2019) is a neural news

[1] https://www.sogou.com/

Table 2: Top-n recommendation performance.

	Model	nDCG@5	nDCG@10	Hit@5	Hit@10
Id-based	BPRMF [33]	0.3011 ± 0.0006	0.3432 ± 0.0002	0.4685 ± 0.0014	0.6053 ± 0.0011
	GRU4Rec [13]	0.3586 ± 0.0005	0.3967 ± 0.0006	0.5218 ± 0.0009	0.6437 ± 0.0010
Content-based	NFM [12]	0.3496 ± 0.0013	0.3989 ± 0.0012	0.5298 ± 0.0023	0.6950 ± 0.0020
	Wide&Deep [2]	0.3589 ± 0.0007	0.4062 ± 0.0004	0.5373 ± 0.0010	0.6938 ± 0.0022
Hybrid	ACCM [35]	0.3885 ± 0.0028	0.4335 ± 0.0031	0.5770 ± 0.0034	0.7258 ± 0.0019
News Rec	NPA [45]	0.4077 ± 0.0021	0.4553 ± 0.0017	0.5979 ± 0.0023	0.7460 ± 0.0008
	NRHUB [46]	0.4132 ± 0.0012	0.4603 ± 0.0012	0.6056 ± 0.0014	*0.7523 ± 0.0018*
	GNewsRec [15]	*0.4236 ± 0.0023*	*0.4674 ± 0.0013*	*0.6122 ± 0.0016*	0.7478 ± 0.0026
Ours	WG4Rec	**0.4459 ± 0.0027****	**0.4908 ± 0.0024****	**0.6379 ± 0.0034****	**0.7750 ± 0.0022****
	WG4Rec (WordNet)	0.4347 ± 0.0024	0.4790 ± 0.0026	0.6282 ± 0.0024	0.7633 ± 0.0034

**. Significantly better than the best baseline (italic ones with underline) with $p < 0.05$ (the same for the following tables).

recommendation model with heterogeneous user behaviors on various domains [46]; **GNewsRec** (2020) is a state-of-the-art GNN-based news recommendation method, in which a heterogeneous user-news-topic graph is constructed to model both long-term and short-term user interests [15].

4.2 Click Prediction

Some previous recommendation methods focus on the click prediction task, whose target is to predict whether user will click a given news document. In this work, WG4Rec is also evaluated on the click prediction task on the public Adressa dataset [9] [2]. The dataset is a news dataset that includes news articles (in Norwegian) in connection with anonymized users. Note that the dataset does not contain cross-scenario content. Some detailed information is shown in Table 3.

Table 3: Statistics of the Adressa dataset and word graph (WG). $\bar{l}$ denotes the average title length.

# news clicks	2,107,312	# users	537,629
# news	14,732	# entity-types	11
# average words $\bar{l}$	4.03	# WG nodes	116,603
# average entities	22.11	# WG edges	13,760,328

We follow the experimental settings of previous work [15] and compare WG4Rec (on single scenario without User Preference Attention) with the reported results. The related news entities are concatenated after news titles. Randomly 20% samples from the last day are for validation, and 80% are for the test set. Negative samples are 1 : 1 randomly sampled from users' unobserved reading history.

4.3 Parameters and Running Environment

All the models, including baselines, are trained with Adam [23] in mini-batches at the size of 128. The learning rate is searched between 1×10^{-6} to 1×10^{-3}, and early-stopping is conducted according to the performance on the validation set. Models are trained at most 100 epochs. The weight of ℓ_2-regularization λ_Θ is searched between 1×10^{-5} to 1×10^{-2} and dropout ratio is set to 0.2 to 0.5. Vector sizes d of all the user, item, and feature vectors are 64. In WG4Rec, at most $n = 100$ neighbors are considered for each relation, and $k = 32$ neighbor words are sampled every batch for $T = 1$ layer GraphSAGE aggregation. The activation function σ is *LeakyRelu* [28]. All models are trained with a GPU (NVIDIA GeForce GTX 2080Ti) with 11GB GPU memory. Each run (including training and testing) of all models can finish in one day.

5 EXPERIMENTAL RESULTS AND ANALYSIS

Experiments are conducted to answer the following questions.

RQ1: How does WG4Rec compare to state-of-the-art news recommendation models in top-n recommendation and click prediction tasks?

RQ2: How does WG4Rec perform on cold-start users?

RQ3: Do different word relations and the word graph improve news recommendation performance?

5.1 Overall Performance (RQ1)

The top-n recommendation performance is shown in Table 2. From the results, the following observations can be concluded. Firstly, id-based methods, i.e., BPRMF and GRU4Rec, perform the worst, showing that content modeling is essential in the news recommendation scenario. Besides, GRU4Rec performs better than BPRMF by modeling the sequential information of user interactions. Secondly, content-based methods, including NFM and Wide&Deep, are better than id-based methods by utilizing text features. Combining CF and CB, ACCM significantly outperforms Wide&Deep and provides comparable performance even with recent news recommendation models. Thirdly, it shows that recent news recommendation models are much better than general content-based models. NRHUB exploits heterogeneous user behaviors, and in each scenario, it uses a similar CNN-based structure with NPA to model the textual content. The performance of NRHUB is better than NPA because it models users' clicks in SERP, which provide more information about personalized preferences. GNewsRec outperforms NPA and NRHUB by taking advantage of graph neural networks to learn user and news representations, which encode high-order structure information by propagating embeddings over the graph. Finally, WG4Rec significantly outperforms all of the baselines. The word graph explicitly models the relationships among words, including *semantically-similar*, *co-occurrence*, and *co-click*, thus providing better word representations for news recommendation. It naturally links cross-scenario content to model users' preferences. We also

[2] http://reclab.idi.ntnu.no/dataset/

Table 4: Click prediction performance on Adressa (Norwegian news, without cross-scenario content). Results of baselines are from previous work [15].

	Model	AUC
Id-based	DMF [50]	0.5566 ± 0.0084
Content-based	LibFM [32]	0.6120 ± 0.0129
	CNN [22]	0.6759 ± 0.0094
	DSSM [17]	0.6861 ± 0.0102
	Wide&Deep [2]	0.6825 ± 0.0112
	DeepFM [10]	0.6909 ± 0.0145
News Rec	DKN [41]	0.7557 ± 0.0113
	DAN [54]	0.7593 ± 0.0125
	GNewsRec [15]	_0.8116 ± 0.0119_
Ours	WG4Rec \ SERP-click	**0.8547 ± 0.0014****

tried replacing the word graph with the synonym graph of WordNet, as shown in the last line of Table 2, and it is significantly worse than the word graph. The reason may be that WordNet is not specifically designed for recommendation and is more sparse.

Some previous methods focus on the click prediction task. Following settings of recent news recommendation work [15], WG4Rec is also tested on their public dataset. Results are in Table 4. Similar observations can be concluded as on the top-n recommendation task. Content-based methods are significantly better than id-based methods, and methods specifically designed for news recommendation perform better than general content-based models. Although recent GNN-based GNewsRec achieves significant improvements, WG4Rec outperforms it significantly as the word graph helps link words and documents to provide better representations.

5.2 Cold-Start and Ablation Performance (RQ2)

Long-tail users take up a large part in real-world recommender systems, and new users are coming to systems every day. We further evaluate the top-n recommendation performance of WG4Rec and baselines on three groups of users based on the number of clicked news in the training set, as shown in Table 5.

Table 5: Top-n recommendation performance (nDCG@10) on cold-start users.

# Clicked News	= 0	= 1, 2, 3	> 3
NPA [45]	-	0.3765	0.4584
NRHUB [46]	_0.2636_	_0.3920_	0.4668
GNewsRec [15]	-	0.3852	_0.4751_
WG4Rec	**0.2982****	**0.4076****	**0.4998****
WG4Rec \ WG	0.2869	0.3985	0.4887
WG4Rec \ SERP-click	-	0.3983	0.4908
WG4Rec \ WG&SERP-click	-	0.3859	0.4869

To investigate the effects of various parts of WG4Rec, we also show WG4Rec without the word graph, clicks in SERP, or neither. Note that models without cross-scenario content cannot work on new users because their user representations are empty, in which condition no personalized information is provided for the recommendation. WG4Rec and NRHUB give predictions on new users by modeling their clicks in SERP, which also draw portraits of users. Firstly, GNewsRec performs better than NRHUB on relatively warm conditions with more than three previous clicked news because GNN better models the context of clicks based on the user-item graph. Secondly, such effects decrease on relatively cold users with less previous clicked news. In contrast, NRHUB performs better than GNewsRec in such a cold scenario by modeling the content information from SERP-clicks. Finally, WG4Rec works the best on all three conditions thanks to modeling the word graph and cross-scenario content. Modeling cross-scenario textual content explores users' content preferences, which is especially valuable for cold users. Nevertheless, WG4Rec without modeling clicks in SERP and only based on news content interactions still outperforms all baselines, verifying that the word graph is an intuitive and effective design to model the relationships between words and news documents. WG4Rec without the word graph is significantly worse than the full WG4Rec, verifying that simply introduce textual content and interactions from other scenarios is not good enough to capture users' preferences as associations between cross-scenario textual content are not well modeled.

In conclusion, WG4Rec handles cold-start users better than previous methods. The word graph and cross-scenario interactions help find related content for both warm and cold users. Textual content from other scenarios, such as clicked content in SERP, enriches the word graph, and in contrast, the word graph improves content modeling in various scenarios. The word graph explicitly models the relationships among words and naturally links textual content and interactions on various scenarios.

5.3 Word Graph Ablation (RQ3)

To investigate the effects of various word relations, we further remove different relations from the word graph. The performances are shown in Figure 4.

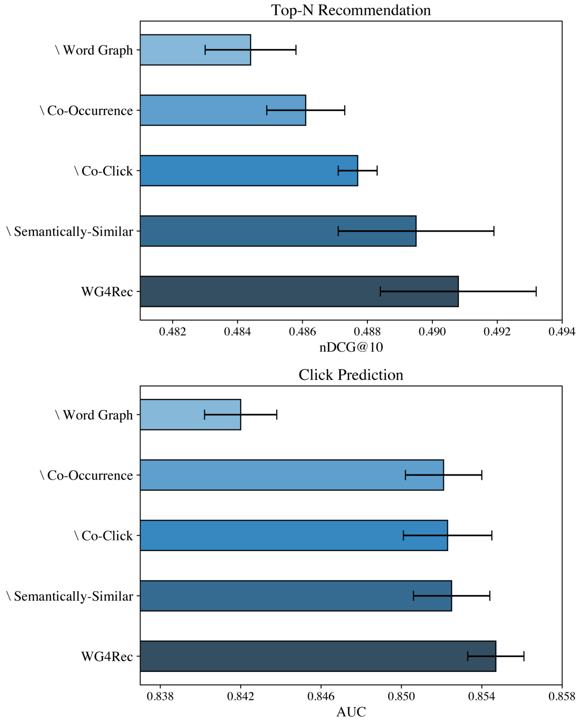


Figure 4: Impacts of removing different relations.

From the results, on both of the top-n recommendation and click prediction tasks, similar observations can be concluded. All three types of relations help enrich the capacity of word graph. *Semantically-similar* and *co-occurrence* words help find similar and related things with users' clicked content. *Semantically-similar* words contribute the least among the three types of relations, verifying that word embeddings are not enough to capture associations among words and documents. Nevertheless, explicitly modeling of *semantically-similar* relations still helps. *Co-occurrence* words provide associations among related words, which are more important to explore users' preferences and not restricted to *semantically-similar* things. *Co-clicks* are not limited to semantic relationships but conduct CF on word-level. It finds the potential interests of users according to the interaction history of other users. They all help improve the effectiveness and generalization of the word graph in various scenarios.

5.4 Online Application for Cold Users (RQ2)

Online recommender systems usually have multiple stages, such as recalling a bunch of documents with some simple strategies and then applying precisely ranking methods. In this work, we also conduct some trials for online applications in the recall step on the platform where the top-n recommendation dataset is collected. We randomly sample some long-tail inactive users who click less than three news documents in one week. They are randomly split into groups for the online A/B test. The control group recalls documents by matching users' interacted keywords. In the experimental group, the word graph is applied to expand their keywords list to 100 words. Other settings of the two groups, including ranking strategies, are the same. We then observe the two groups for two weeks, and some statistics are shown in Table 6. Note that the actual values are scaled at a certain ratio due to the company's requirements, but the relative improvements are precise.

Table 6: Results of online A/B test. The presented results are scaled values of actual statistics but the relative improvements are precise.

	CTR	Avg. Clicked News
Base	0.0905	0.4080
Word Graph	0.1148	0.4964
Improvements	*+26.89%*	*+21.67%*

It is encouraging that the click-through rate (CTR) has a 26.9% promotion relatively compared to using the original keywords list for this part of user traffic. Besides, the average number of clicked news per user has a relative improvement of 21.67%. It is valuable for systems to attract and retain these long-tail users. It verifies that the word graph models the associations of words and documents specifically for personalized recommendation and helps retrieve some related news documents precisely for each user.

5.5 Case Study

Some real cases in different word relations are shown in Table 7. *Semantically-similar* words of "monitor" and "oil" have close meanings, and those of the word "teeth" are related oral parts.

Table 7: Examples of top words with different relations.

Central	*Semantically-Similar*	*Co-Occurrence*	*Co-Click*
monitor	liquid-crystal LCD display	computer gaming host	mainboard Asus graphics
teeth	gum tartar gingival	dentist health whiten	calculus OCD blackhead
oil	oilman crude petroleum	Saudi China price	U.S. country Trump

Co-occurrence words are semantically related. For example, "monitor" often refers to "computer monitor", and the "gaming monitor" is a kind of monitor with higher refresh rate. "Dentists" take care of our "teeth health", and sometimes people may need a "teeth whitenin"g. "Saudi Arabia" is one of the world's largest "oilmen", and "China" is one of the world's largest "oil consumer", both of which are sensitive to the "oil price".

Co-click words indicate potential interests of users that may not be so obvious. For example, one clicked news about "monitors" may also be interested in other computer hardware such as the "mainboard" or "graphics" card. One interested in "teeth whitening" may also want to clean dental "calculus" or "blackheads" for beauty. A user who likes reading news about the "oil" market may prefer financial and political topics where the "U.S." and "Trump" are frequently mentioned. These words in three relations all help model the central words and help understand and explore users' preferences.

6 CONCLUSIONS

This paper proposes a news recommendation framework named WG4Rec, which uses a word graph and utilizes cross-scenario information for recommendation. The construction of word graph does not rely on external knowledge and can be offline constructed based on cross-scenario content and interactions. We extract three types of word relations, including *semantically-similar*, *co-occurrence*, and *co-click*. *Semantically-similar* words bring additional information about similar content in terms of semantics to users. *Co-occurrence* words explore users' interests in the same document context. *Co-click* words enable word-level CF across documents. Based on the word graph, WG4Rec uses a multi-level and multi-scenario structure to model user preferences for the news recommendation. Experimental results show that WG4Rec outperforms state-of-the-art news recommendation models on both top-n recommendation and click prediction tasks on both warm and cold users. It verifies explicitly modeling word relations help link textual content and explore user interests. On our top-n recommendation dataset, WG4Rec utilizes users' clicks in SERP to enrich the word graph and better understand users. Results verify that search logs are valuable supplements to recommender systems. The online A/B test also shows encouraging results by applying the word graph.

In the future, we plan to explore more relationships among words to enrich the word graph (e.g., combination with knowledge graph) and boost the GNN modules in WG4Rec.

REFERENCES

[1] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An Efficient Adaptive Transfer Neural Network for Social-aware Recommendation. In *SIGIR*. ACM, 225–234.

[2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *DLRS@RecSys*. ACM, 7–10.

[3] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*. ACL, 1724–1734.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4171–4186.

[5] Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing Word Lattice Translation. In *ACL*. The Association for Computer Linguistics, 1012–1020.

[6] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In *WWW*. ACM, 278–288.

[7] Christiane Fellbaum. 2012. WordNet. *The encyclopedia of applied linguistics* (2012).

[8] Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph Enhanced Representation Learning for News Recommendation. In *WWW*. ACM / IW3C2, 2863–2869.

[9] Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. 2017. The Adressa dataset for news recommendation. In *WI*. ACM, 1042–1048.

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *IJCAI*. ijcai.org, 1725–1731.

[11] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*. 1024–1034.

[12] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In *SIGIR*. ACM, 355–364.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR (Poster)*.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (1997), 1735–1780.

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

[16] Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph Neural News Recommendation with Unsupervised Preference Disentanglement. In *ACL*. Association for Computational Linguistics, 4255–4264.

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

[18] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*. ACM, 135–142.

[19] Mingi Ji, Weonyoung Joo, Kyungwoo Song, Yoon-Yeong Kim, and Il-Chul Moon. 2020. Sequential Recommendation with Relation-Aware Kernelized Self-Attention. In *AAAI*. AAAI Press, 4304–4311.

[20] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P. Xing. 2019. Rethinking Knowledge Graph Propagation for Zero-Shot Learning. In *CVPR*. Computer Vision Foundation / IEEE, 11487–11496.

[21] Dhruv Khattar, Vaibhav Kumar, Vasudeva Varma, and Manish Gupta. 2018. Weave&Rec: A Word Embedding based 3-D Convolutional Network for News Recommendation. In *CIKM*. ACM, 1855–1858.

[22] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*. ACL, 1746–1751.

[23] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

[24] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*. OpenReview.net.

[25] Vaibhav Kumar, Dhruv Khattar, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Neural Architecture for News Recommendation. In *CLEF (Working Notes) (CEUR Workshop Proceedings, Vol. 1866)*. CEUR-WS.org.

[26] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *CIKM*. ACM, 1419–1428.

[27] Weizhi Ma, Min Zhang, Chenyang Wang, Cheng Luo, Yiqun Liu, and Shaoping Ma. 2018. Your Tweets Reveal What You Like: Introducing Cross-media Content Information into Multi-domain Recommendation. In *IJCAI*. ijcai.org, 3484–3490.

[28] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, Vol. 30. 3.

[29] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based News Recommendation for Millions of Users. In *KDD*. ACM, 1933–1942.

[30] Stefan Ortmanns, Hermann Ney, and Xavier L. Aubert. 1997. A word graph algorithm for large vocabulary continuous speech recognition. *Comput. Speech Lang.* 11, 1 (1997), 43–72.

[31] Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web* 8, 3 (2017), 489–508.

[32] Steffen Rendle. 2012. Factorization Machines with libFM. *ACM Trans. Intell. Syst. Technol.* 3, 3 (2012), 57:1–57:22.

[33] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. AUAI Press, 452–461.

[34] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC (Lecture Notes in Computer Science, Vol. 10843)*. Springer, 593–607.

[35] Shaoyun Shi, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Attention-based Adaptive Model to Unify Warm and Cold Starts Recommendation. In *CIKM*. ACM, 127–136.

[36] Amit Singhal. 2012. Introducing the knowledge graph: things, not strings. *Official google blog* 5 (2012).

[37] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *AAAI*. AAAI Press, 4444–4451.

[38] Roy Tromble, Shankar Kumar, Franz Josef Och, and Wolfgang Macherey. 2008. Lattice Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *EMNLP*. ACL, 620–629.

[39] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing Cold Start in Recommender Systems. In *NIPS*. 4957–4966.

[40] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In *CIKM*. ACM, 417–426.

[41] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *WWW*. ACM, 1835–1844.

[42] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* 29, 12 (2017), 2724–2743.

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

[44] Frank Wessel, Ralf Schlüter, Klaus Macherey, and Hermann Ney. 2001. Confidence measures for large vocabulary continuous speech recognition. *IEEE Trans. Speech Audio Process.* 9, 3 (2001), 288–298.

[45] 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.

[46] Chuhan Wu, Fangzhao Wu, Mingxiao An, Tao Qi, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Heterogeneous User Behavior. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 4873–4882.

[47] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Multi-Head Self-Attention. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 6388–6393.

[48] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Networks Learn. Syst.* 32, 1 (2021), 4–24.

[49] Yang Xu, Lei Zhu, Zhiyong Cheng, Jingjing Li, and Jiande Sun. 2020. Multi-Feature Discrete Collaborative Filtering for Fast Cold-Start Recommendation. In *AAAI*. AAAI Press, 270–278.

[50] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems. In *IJCAI*. ijcai.org, 3203–3209.

[51] Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. 2019. A Vectorized Relational Graph Convolutional Network for Multi-Relational Network Alignment. In *IJCAI*. ijcai.org, 4135–4141.

[52] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[53] Tong Zhao, Julian J. McAuley, and Irwin King. 2014. Leveraging Social Connections to Improve Personalized Ranking for Collaborative Filtering. In *CIKM*. ACM, 261–270.

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