Personalized News Recommendation with Knowledge-aware Interactive Learning

Anonymous Author(s)

ABSTRACT

Personalized news recommendation is essential for online news platforms to help users find their interested news. Existing methods usually first represent news based on their content and then recommend news based on the relevance between candidate news and users' interest inferred from their clicked news. These methods model clicked news and candidate news independently, where the fine-grained relatedness between them cannot be effectively captured. In addition, the relatedness between entities of news on the knowledge graph, which can provide rich clues for modeling the relatedness between news, cannot be effectively exploited by existing methods. In this paper, we propose a personalized news recommendation framework with knowledge-aware interactive learning, which can effectively model the fine-grained relatedness between users' clicked news and candidate news. We propose an interactive knowledge encoder to capture the fine-grained interactions between the entities of clicked news and candidate news with the help of knowledge graphs. In addition, we propose an interactive text encoder to model the fine-grained interactions between the contexts of clicked news and candidate news. Besides, since the interest of a user may be diverse and different users may be interested in different aspects of news, we propose an interactive user-candidate encoder to learn candidate-aware user representation and personalized candidate news representation to better model the relations between user interest and the content of candidate news. Extensive experiments on two real-world datasets show that our method can effectively improve the performance of news recommendation and outperform many baseline methods.

CCS CONCEPTS

• **Information systems → Personalization;Recommender systems**;

KEYWORDS

Personalized News Recommendation, Knowledge Graph, Interactive Learning

ACM Reference Format:

Anonymous Author(s). 2021. Personalized News Recommendation with Knowledge-aware Interactive Learning. In *Proceedings of The Web Conference (WWW 2021).* ACM, New York, NY, USA, Article 4, 11 pages. https://doi.org/10.475/123_4

ACM ISBN 123-4567-24-567/08/06.

https://doi.org/10.475/123_4

1 INTRODUCTION

In recent years, online news platforms such as Yahoo! News[1] and Bing News[2], have attracted a huge number of users to consume online news information [22, 33]. However, since massive news articles are generated and collected by these platforms every day, users usually have difficulties in finding the news information they need. Personalized news recommendation techniques, which aim to help users find their interested news, usually play an essential role in online news platforms to alleviate the information overload of users. Thus, the study on personalized news recommendation has attracted much attention from both academia and industry [1, 2, 12, 35, 37, 39].

Most existing methods for personalized news recommendation usually first model news articles from their content, and then recommend news based on the relevance between candidate news and users' interest inferred from their news click history [22]. For example, Okura et al. [22] proposed to apply an auto-encoder to learn news content representation from news bodies and apply a GRU network to learn user interest representation from their clicked news. In addition, since entities in news can provide rich clues for understanding news content, many methods incorporate entities in news texts to enhance news modeling [19, 32]. For instance, Wang et al. [32] proposed a knowledge-aware convolutional network to learn news representations from words and entities in news titles and utilized a candidate-aware attention network to learn user interest representations from their clicked news. However, these methods usually model the content of clicked news and candidate news independently and do not consider the relatedness between them when learning their representations, which may not be optimal for further measuring the relevance between candidate news and the user interest inferred from clicked news.

In fact, the clicked news of users and candidate news may have some fine-grained relatedness in their contexts and entities. For example, as shown in Fig. 1, the contexts in the user's clicked news such as "trending song" and the contexts in candidate news such as "popular movie" have some inherent relatedness, which can provide useful clues for understanding that this user may be interested in popular works. In addition, the entity "Style" in clicked news has relatedness with the entity "Movie Cats" in candidate news since the former is the song of the Taylor and the chief actress of the cats movie is also Taylor. The fine-grained relatedness between entities can help infer the user interest in the actress Taylor. Thus, capturing these fine-grained relations between user's clicked news and candidate news is beneficial for modeling the relevance between user interest and news content in news recommendation.

In this paper, we propose a personalized news recommendation framework with knowledge-aware interactive learning (named KAIL), which can effectively model the fine-grained relatedness

[1] https://news.yahoo.com
[2] https://www.bing.com/news

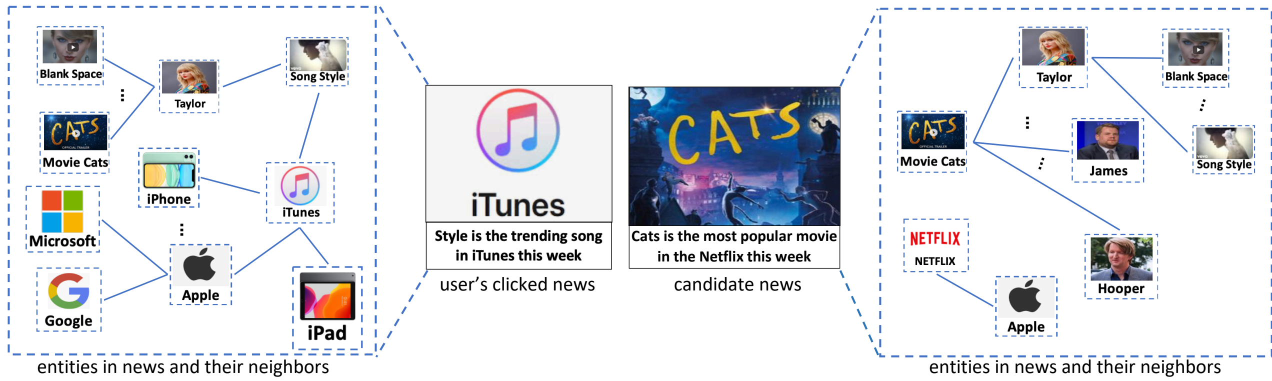


Figure 1: An example user's clicked news and candidate news with their entities on the knowledge graph.

between users' clicked news and candidate news. In the framework, we propose an interactive knowledge encoder to capture fine-grained interactions between the entities of clicked news and candidate news. Specifically, we first propose a graph co-attention network to learn representations of entities by selecting their neighbors which are informative for modeling the relatedness between entities in user's clicked news and candidate news. We further propose to use an entity co-attention network to learn knowledge representations for user's clicked news and candidate news by capturing fine-grained interactions between their entities. Moreover, we also propose an interactive text encoder to learn text representations for user's clicked news and candidate news by capturing fine-grained interactions between their contexts. The unified representation of news is formulated as the aggregation of its knowledge representation and text representation. Since the interest of a user may be diverse and users may focus on different aspects of the content of candidate news, we further use an interactive user-candidate encoder to learn candidate-aware user representation and personalized candidate news representation to better capture user interest in news content. Finally, the candidate news is ranked based on the relevance between the representations of candidate news and user interest. We conduct extensive experiments on two real-world datasets. The experimental results show that our method can effectively improve the performance of news recommendation and outperform other baseline methods.

2 RELATED WORK

Personalized news recommendation is a key technique to alleviate information overload in online news platforms, which has been widely studied in recent years [6, 15–18, 20, 30]. These methods usually first represent news based on their content and then recommend news based on the relevance between news content and user interest learned from their reading history [32, 33, 36]. For example, Okura et al. [22] designed a de-noising auto-encoder to learn news representations from news bodies and unitized a GRU network to learn user representation from their clicked news. They further performed dot product between representations of user interest and candidate news to measure their relevance. Wu et al. [33] proposed to learn news representations from news titles, bodies, categories as well as sub-categories via an attentive multi-view learning framework and applied an attention network to learn user interest representation from representations of user's clicked news. Wu et al. [36] proposed to use multi-head self-attention networks to learn news representations by capturing the relatedness among words in news titles and learn user representation by modeling the relatedness between user's clicked news. In addition, these methods also used dot product to compute the relevance score between user interest and candidate news. Besides plain news texts, the entities in news are also used to enhance news content modeling [4, 7, 19, 32]. For example, Chu et al. [4] proposed to learn news representations based on the aggregation of the average embeddings of words in news and average embeddings of entities in news. Wang et al. [32] proposed a knowledge-aware convolutional network to learn news representations from words and entities in news titles. Liu et al. [19] proposed to use a graph attention network to learn knowledge representations from entities in news and their neighbors on the knowledge graph. However, these methods only model the relatedness between user's clicked news and candidate news based on their representations which are learned independently, and they cannot capture the fine-grained relatedness between them when learning their representations. Only a few methods consider the fine-grained relatedness between clicked news and candidate news [31]. For example, Wang et al. [31] proposed to capture fine-grained user interest in candidate news based on the similarities between words in user's clicked news and candidate news. However, their method only models the fine-grained relatedness between clicked news and candidate news from their texts, but do not consider the useful entities in news. Different from these methods, our *KAIL* method leverages a knowledge-aware interactive learning methods to model the interactions between candidate news and users' clicked news in terms of their contexts and entities, which can effectively mine the fine-grained relatedness between them to facilitate news recommendation.

3 METHODOLOGY

In this section, we first introduce the problem definition of personalized news recommendation. Next, we introduce our personalized news recommendation method with knowledge-aware interactive news representation learning (named *KAIL*), which can effectively capture the fine-grained relatedness between user's clicked news

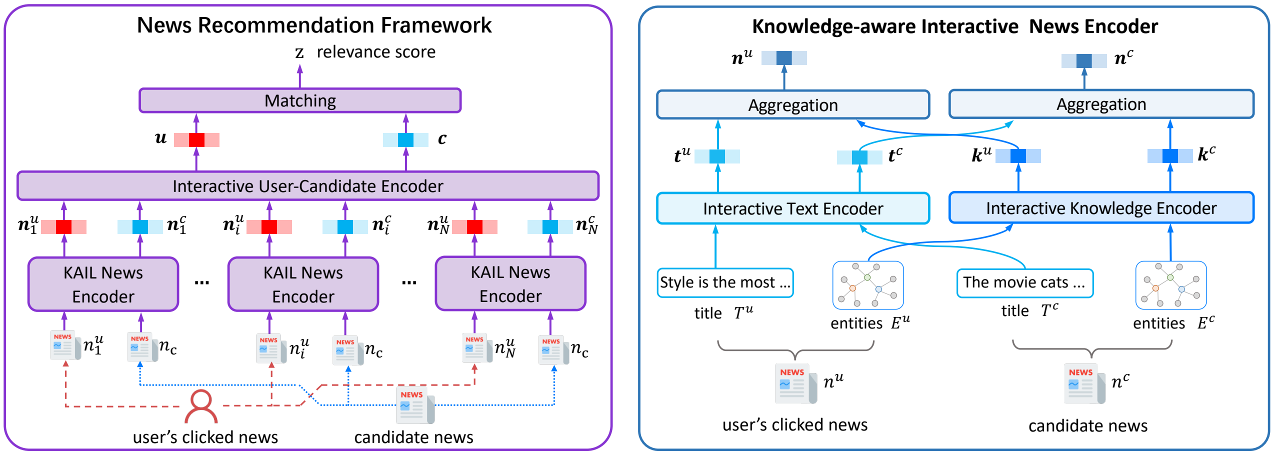


Figure 2: The news recommendation framework and knowledge-aware interactive news encoder of the *KAIL* method.

and candidate news. We first introduce the news recommendation framework of *KAIL*. Then we introduce the architecture of each module in *KAIL*.

3.1 Problem Formulation

In this section, we formulate the personalized news recommendation problem as follows. Given a user u and a candidate news n^c, we need to compute the relevance score z measuring the interest of user u in the content of candidate news n^c. Then different candidate news are ranked and recommended to user u based on their relevance scores. The user u is associated with the set of his/her clicked news. Each news n is associated with its title T and the entities E. Besides, there is a knowledge graph $\mathcal{G}$ used to provide the relatedness between entities, and each entity e in $\mathcal{G}$ is associated with its pre-trained embeddings $\mathbf{e}$.

3.2 News Recommendation Framework

In this section, we introduce the news recommendation framework of *KAIL*. As illustrated in Fig. 2, there are two major modules in *KAIL*. The first one is a *knowledge-aware interactive news encoder*, which learns the knowledge-aware interactive representations of a user's clicked news and the candidate news by capturing the fine-grained interactions between their entities and between their titles. The second one is an *interactive user-candidate encoder*, which learns candidate-aware user interest representation and personalized candidate news representation from the interactive representations of user's clicked news and candidate news generated by the *knowledge-aware interactive news encoder*. Finally, we measure user interest in news content based on the relevance between the candidate-aware user interest representation and personalized candidate news representation. Next, we introduce each module in detail.

3.3 Knowledge-aware Interactive News Encoder

In this section, we introduce the framework of the *knowledge-aware interactive news encoder* (shorted as *KAIL news encoder*), which learns representations of a user's clicked news n^u and candidate news n^c from their titles and entities. As shown in Fig. 2, it contains three sub-modules. The first one is an *interactive knowledge encoder* (denoted as ϕ_k), which learns interactive knowledge representations $\mathbf{k}^u \in \mathcal{R}^{d_k}$ and $\mathbf{k}^c \in \mathcal{R}^{d_k}$ for news n^u and news n^c by capturing the fine-grained interactions between their entities based on the knowledge graph:

$$[\mathbf{k}^u, \mathbf{k}^c] = \phi_k(E^u, E^c), \tag{1}$$

where d_k denotes knowledge representation dimensions, E^u and E^c denote entities in news n^u and n^c respectively. The second one is an *interactive text encoder* (denoted as ϕ_t), which learns interactive text representations $\mathbf{t}^u \in \mathcal{R}^{d_t}$ and $\mathbf{t}^c \in \mathcal{R}^{d_t}$ for news n^u and n^c by capturing the fine-grained interactions between their titles:

$$[\mathbf{t}^u, \mathbf{t}^c] = \phi_t(T^u, T^c), \tag{2}$$

where d_t denotes text representation dimensions, T^u and T^c denote titles of news n^u and n^c respectively. Finally, we project the interactive knowledge representation and text representation of the same news to learn the unified news representation:

$$\mathbf{n}^u = \mathbf{P}_n[\mathbf{t}^u; \mathbf{k}^u], \qquad \mathbf{n}^c = \mathbf{P}_n[\mathbf{t}^c; \mathbf{k}^c], \tag{3}$$

where $\mathbf{n}^u \in \mathcal{R}^{d_n}$ denotes the knowledge-aware interactive representation of user's clicked news n^u, $\mathbf{n}^c \in \mathcal{R}^{d_n}$ denotes the corresponding knowledge-aware interactive representation of candidate news n^c, d_n denotes news representation dimensions, $[\cdot;\cdot]$ denotes the concatenation operation, and $\mathbf{P}_n \in \mathcal{R}^{d_n \times (d_t + d_k)}$ denotes the trainable projection matrix.

3.4 Interactive Knowledge Encoder

In this section, we introduce the architecture of the *interactive knowledge encoder*, which learns the interactive knowledge representation of user's clicked news n^u and candidate news n^c from fine-grained interactions between their entities E^u and E^n based on the knowledge graph $\mathcal{G}$. As shown in Fig. 3, it can be decomposed into three components. The first one is a stacked graph attention (GAT) network [28]. To summarize the information for each entity

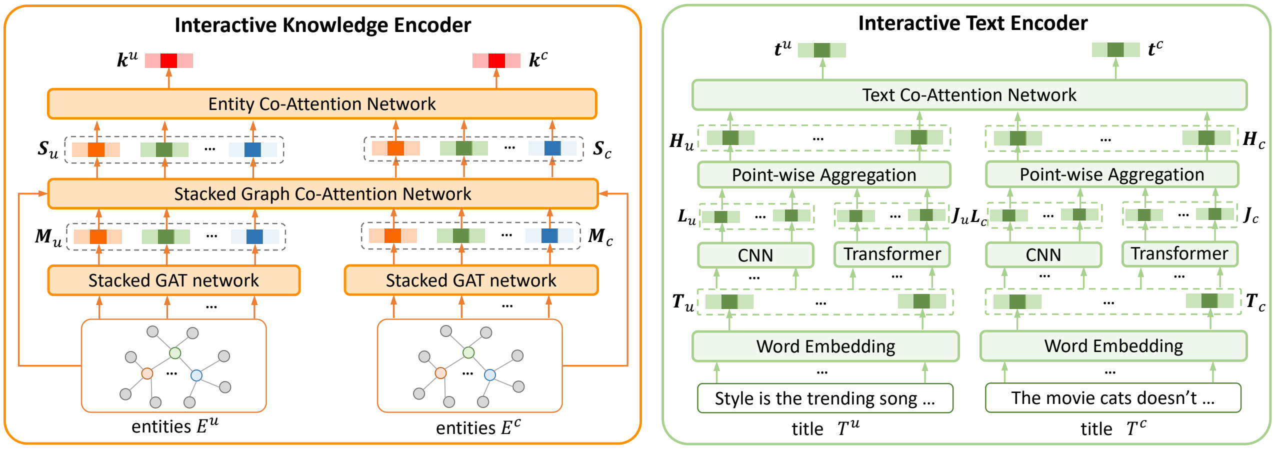


Figure 3: The architecture of the interactive knowledge encoder and interactive text encoder.

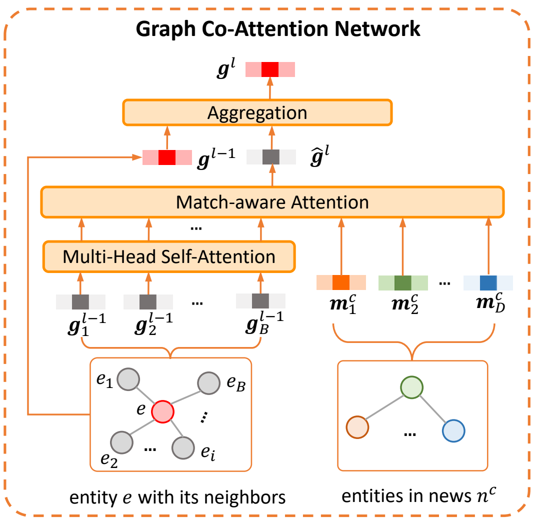


Figure 4: The architecture of the GCAT network.

in E^u or E^c from their neighbors within K hops, we utilize a GAT network [28] stacked K layers to learn their representations, which are denoted as $\mathbf{M}_u = \{\mathbf{m}_i^u\}_{i=1}^D \in \mathcal{R}^{d_k \times D}$ and $\mathbf{M}_c = \{\mathbf{m}_i^c\}_{i=1}^D \in \mathcal{R}^{d_k \times D}$ respectively, where D denotes number of entities in news.

The second one is a stacked graph co-attention (GCAT) network proposed in this paper. Note that an entity usually has rich relatedness with different entities on the knowledge graph [8, 29] and informative relatedness usually has different importance for modeling the fine-grained relatedness between different news. For example, as show in Fig. 1, the entity "Cats" has many neighbor entities, such as its director "James", chief actor "Hooper", chief actress "Taylor" and so on, while only entity "Taylor" is informative for modeling the relatedness between user's clicked news and candidate news since it is also the neighbor of entity "Style" in user's clicked news. To better select informative relatedness between entities for modeling the fine-grained relatedness between user's clicked news and candidate news, we propose a graph co-attention network (GCAT) stacked K layers to learn match-aware representations for entities in news n^u and n^c. Take an entity e in news n^u as example, the l-th graph co-attention network shown in Fig. 4 learns its representation by aggregating its neighbors with the supervision of entities in news n^c. Specifically, we first apply a multi-head self-attention network [27] to the representations of its neighbor entities generated by the $l-1$-th GCAT network[3] to model the relatedness between different neighbor entities. Next, we propose a match-aware attention network to aggregate neighbor entities of entity e by capturing their relevance with entities in news n^c, which is measured by a relevance matrix $\mathbf{I}_u \in \mathcal{R}^{D \times B}$:

$$\mathbf{I}_u = \mathbf{M}_c^T \mathbf{W}_c^c \hat{\mathbf{G}}_l, \tag{4}$$

where $\hat{\mathbf{G}}_l = \{\hat{\mathbf{g}}_i^l\}_{i=1}^B \in \mathcal{R}^{d_k \times B}$ denotes representations of neighbor entities generated by the self-attention network, B denotes the number of neighbors, and $\mathbf{W}_c^c \in \mathcal{R}^{d_k \times d_k}$ is trainable weights. Then the attention vector $\mathbf{v}^u \in \mathcal{R}^B$ of neighbor entities is calculated based on the relevance matrix:

$$\mathbf{v}^u = \mathbf{q}_e^T \cdot \tanh(\mathbf{W}_s^c \hat{\mathbf{G}}^l + \mathbf{W}_h^c \mathbf{M}_c f(\mathbf{I}_u)), \tag{5}$$

where f denotes the softmax activation which normalizes each column vector of the input matrix, $\mathbf{q}_e \in \mathcal{R}^{d_q}$ denotes the trainable attention query, d_q denotes its dimensions, $\mathbf{W}_s^c \in \mathcal{R}^{d_q \times d_k}$ and $\mathbf{W}_h^c \in \mathcal{R}^{d_q \times d_k}$ are trainable weights. Then we aggregates neighbors of entity e into a unified representation $\hat{\mathbf{g}}^l \in \mathcal{R}^{d_k}$:

$$\hat{\mathbf{g}}^l = \sum_{i=1}^{B} \lambda_i^u \hat{\mathbf{g}}_i^l, \qquad \lambda_i^u = \frac{\exp(v_i^u)}{\sum_{j=1}^{B} \exp(v_j^u)} \tag{6}$$

where λ_i^u denotes the attention weight of the i-th neighbor entity. Finally the representation $\mathbf{g}^l \in \mathcal{R}^{d_k}$ of the entity e generated by the l-th GCAT network is formulated as:

$$\mathbf{g}^l = \mathbf{P}_e[\hat{\mathbf{g}}^l; \mathbf{g}^{l-1}], \tag{7}$$

[3] The input of the 1-th GCAT network are the initialized embeddings of each entity.

where $\mathbf{P}_e \in \mathcal{R}^{d_k \times 2d_k}$ denotes the projection matrix. In this way, the GCAT network stacked K layers can learn match-aware representations $\mathbf{S}_u = \{\mathbf{s}_i^u\}_{i=1}^D \in \mathcal{R}^{d_k \times D}$ for entities in user's clicked news by capturing the relatedness between their neighbors within K hops and entities in candidate news. In a symmetrical way, we can learn the match-aware representations $\mathbf{S}_c = \{\mathbf{s}_i^c\}_{i=1}^D \in \mathcal{R}^{d_k \times D}$ of entities in candidate news.

The third one is an entity co-attention network. Note that entities in news usually have different informativeness for modeling the fine-grained relatedness between user's clicked news and candidate news. For example, as shown in Fig. 1, in user's clicked news, the entity "Style"is more informative than the entity "iTunes" since the entity "Style" has relatedness with the entity "Cats" in candidate news. Thus, we apply an entity co-attention network to learn interactive knowledge representations for news n^u and n^c by capturing fine-grained interactions between their entities to model their fine-grained relatedness. In detail, the affinity matrix $\mathbf{C}_e \in \mathcal{R}^{D \times D}$ measuring the relevance among entities of news n^u and n^c is formulated as:

$$\mathbf{C}_e = \mathbf{S}_c^T \mathbf{W}_c^k \mathbf{S}_u, \tag{8}$$

where $\mathbf{W}_c^k \in \mathcal{R}^{d_k \times d_k}$ is the trainable weights. Then we calculate attention vector $\mathbf{a}^u \in \mathcal{R}^D$ and $\mathbf{a}^c \in \mathcal{R}^D$ of entities in news n^u and n^c respectively:

$$\mathbf{a}^u = \mathbf{q}_k^T \cdot \tanh(\mathbf{W}_s^k \mathbf{S}_u + \mathbf{W}_h^k \mathbf{S}_c f(\mathbf{C}_e)), \tag{9}$$

$$\mathbf{a}^c = \mathbf{q}_k^T \cdot \tanh(\mathbf{W}_s^k \mathbf{S}_c + \mathbf{W}_h^k \mathbf{S}_u f(\mathbf{C}_e^T)), \tag{10}$$

where $\mathbf{q}_k \in \mathcal{R}^{d_q}$ is the trainable attention query, and $\mathbf{W}_s^k \in \mathcal{R}^{d_q \times d_k}$, $\mathbf{W}_h^k \in \mathcal{R}^{d_q \times d_k}$ are trainable weights. Finally interactive knowledge representations $\mathbf{k}^u \in \mathcal{R}^{d_k}$ and $\mathbf{k}^c \in \mathcal{R}^{d_k}$ of user's clicked news and candidate news are formulated as:

$$\mathbf{k}^u = \sum_{i=1}^{D} \alpha_i^u \mathbf{s}_i^u, \qquad \alpha_i^u = \frac{\exp(a_i^u)}{\sum_{j=1}^D \exp(a_j^u)}, \tag{11}$$

$$\mathbf{k}^c = \sum_{i=1}^{D} \alpha_i^c \mathbf{s}_i^c, \qquad \alpha_i^c = \frac{\exp(a_i^c)}{\sum_{j=1}^D \exp(a_j^c)}, \tag{12}$$

where α_i^u and α_i^c denote the attention weight of the i-th entity in news n^u and n^c respectively.

3.5 Interactive Text Encoder

In this section, we introduce the architecture of the *interactive text encoder*, which learns the interactive text representations for user's clicked news n^u and candidate news n^c from their titles T^u and T^c. As shown in Fig. 3, we first independently learn contextual word representations for title T^u and T^c. Specifically, take title T^u as an example, we first covert it into an embedding vector sequence $\mathbf{T}_u \in \mathcal{R}^{d_g \times M}$ via the word embedding layer to enhance the semantic information, where d_g denotes word embedding dimensions, and M denotes the number of words in title. Next, since local contexts are important for text modeling [33], we apply a CNN network [13] to $\mathbf{T}_u$ to learn local-contextual word representations $\mathbf{L}_u \in \mathcal{R}^{d_t \times M}$ by capturing local contexts. In addition, long-range contexts are also informative for understanding news content [36]. Thus we apply the transformer [27] to $\mathbf{T}_u$ to learn global-contextual word representations $\mathbf{J}_u \in \mathcal{R}^{d_t \times M}$ from long-range contexts. Then, we averages the local- and global- contextual representations of each word to learn their unified contextual representations $\mathbf{H}_u = \{\mathbf{h}_i^u\}_{i=1}^M \in \mathcal{R}^{d_t \times M}$ respectively, where $\mathbf{h}_i^u \in \mathcal{R}^{d_t}$ denotes the contextual representation of the i-th word in title T^u. Besides, we can learn contextual word representations $\mathbf{H}_c = \{\mathbf{h}_i^c\}_{i=1}^M \in \mathcal{R}^{d_t \times M}$ for title T^c in the same way, where $\mathbf{h}_i^c \in \mathcal{R}^{d_t}$ denotes the contextual representation of the i-th word in title T^c.

Finally, note that news content may contain different aspects and the critical aspects for modeling the fine-grained relatedness between different news are usually different. For example, given an example news "Apple's plans to make over-ear headphones." containing two aspects, i.e., "Apple's product plan" and "headphones", the former is informative for modeling example's relatedness with news "The best headphones of 2020." since users interested in headphones may click both of them and the latter is informative for modeling example's relatedness with news "iPhone 12 cases buyer's guide." since users interested in the product of Apple may also click these two news. Thus, we apply a text co-attention network [25] to learn interactive text representations of news n^u and n^c by capturing fine-grained interactions between their titles. Specifically, we first calculate the affinity matrix $\mathbf{C}_t \in \mathcal{R}^{M \times M}$ measuring the semantic relevance between different words in title T^u and T^c:

$$\mathbf{C}_t = \mathbf{H}_c^T \mathbf{W}_c^t \mathbf{H}_u, \tag{13}$$

where $\mathbf{W}_c^t \in \mathcal{R}^{d_t \times d_t}$ is the trainable weights. Then we compute the attention vector $\mathbf{b}^u \in \mathcal{R}^M$ and $\mathbf{b}^c \in \mathcal{R}^M$ for words in user's clicked news and candidate news respectively based on $\mathbf{C}_t$:

$$\mathbf{b}^u = \mathbf{q}_t^T \cdot \tanh(\mathbf{W}_s^t \mathbf{H}_u + \mathbf{W}_h^t \mathbf{H}_c f(\mathbf{C}_t)), \tag{14}$$

$$\mathbf{b}^c = \mathbf{q}_t^T \cdot \tanh(\mathbf{W}_s^t \mathbf{H}_c + \mathbf{W}_h^t \mathbf{H}_u f(\mathbf{C}_t^T)), \tag{15}$$

where $\mathbf{q}_t \in \mathcal{R}^{d_q}$ denotes the trainable attention query, $\mathbf{W}_s^t \in \mathcal{R}^{d_q \times d_t}$ and $\mathbf{W}_h^t \in \mathcal{R}^{d_q \times d_t}$ are the trainable parameters. Finally, the interactive text representations $\mathbf{t}^u \in \mathcal{R}^{d_t}$ and $\mathbf{t}^c \in \mathcal{R}^{d_t}$ of news n^u and n^c are formulated as:

$$\mathbf{t}^u = \sum_{i=1}^{M} \beta_i^u \mathbf{h}_i^u, \qquad \beta_i^u = \frac{\exp(b_i^u)}{\sum_{j=1}^M \exp(b_j^u)}, \tag{16}$$

$$\mathbf{t}^c = \sum_{i=1}^{M} \beta_i^c \mathbf{h}_i^c, \qquad \beta_i^c = \frac{\exp(b_i^c)}{\sum_{j=1}^M \exp(b_j^c)}, \tag{17}$$

where β_i^u and β_i^c denotes attention weight of the i-th word in title T^u and T^c respectively.

3.6 Interactive User-Candidate Encoder

In this section, we introduce the detail of *interactive user-candidate encoder*, which learns candidate-aware user interest representation and personalized candidate news representation from user's clicked news and candidate news. Since interests of a user may be diverse [21], different news clicked by the user usually have different informativeness for measuring user interest in the candidate news. Thus learning candidate-aware user interest representation can better model user interest for capturing its relevance with candidate news. Besides, since different users may focus on different aspects of candidate news, learning personalized candidate news representation is also beneficial for modeling user interest in the content of candidate news. Thus, in the *interactive user-candidate encoder*, we apply a news co-attention network to learn candidate-aware user representation and personalized candidate news representation. Specifically, we first calculate the affinity matrix $\mathbf{C}_n \in \mathcal{R}^{N\times N}$ based on the interactive representations of user's clicked news $\mathbf{N}_u = \{\mathbf{n}_i^u\}_{i=1}^N \in \mathcal{R}^{d_n\times N}$ and candidate news $\mathbf{N}_c = \{\mathbf{n}_i^c\}_{i=1}^N \in \mathcal{R}^{d_n\times N}$ to measure their relevance:

$$\mathbf{C}_n = \mathbf{N}_c^T \mathbf{W}_c^n \mathbf{N}_u, \tag{18}$$

where N denotes the number of clicked news, $\mathbf{n}_i^u \in \mathcal{R}^{d_n}$ denotes the interactive representation of user's i-th clicked news, $\mathbf{n}_i^c \in \mathcal{R}^{d_n}$ denotes the corresponding interactive representation of candidate news, and $\mathbf{W}_c^n \in \mathcal{R}^{d_n\times d_n}$ is the trainable weights. Then we compute the attention vector $\mathbf{r}^u \in \mathcal{R}^N$ and $\mathbf{r}^c \in \mathcal{R}^N$ for the representations of user's clicked news and candidate news based on the affinity matrix:

$$\mathbf{r}^u = \mathbf{q}_n^T \cdot \tanh(\mathbf{W}_s^n \mathbf{N}_u + \mathbf{W}_h^n \mathbf{N}_c f(\mathbf{C}_n)), \tag{19}$$

$$\mathbf{r}^c = \mathbf{q}_n^T \cdot \tanh(\mathbf{W}_s^n \mathbf{N}_c + \mathbf{W}_h^n \mathbf{N}_u f(\mathbf{C}_n^T)), \tag{20}$$

where $\mathbf{q}_n \in \mathcal{R}^{d_q}$ denotes the attention query, $\mathbf{W}_s^n \in \mathcal{R}^{d_q\times d_n}$ and $\mathbf{W}_h^n \in \mathcal{R}^{d_q\times d_n}$ are the trainable weights. The candidate-aware user representation $\mathbf{u} \in \mathcal{R}^{d_n}$ and personalized candidate news representation $\mathbf{c} \in \mathcal{R}^{d_n}$ are formulated as:

$$\mathbf{u} = \sum_{i=1}^N \gamma_i^u \mathbf{n}_i^u, \qquad \gamma_i^u = \frac{\exp(r_i^u)}{\sum_{j=1}^N \exp(r_j^u)}, \tag{21}$$

$$\mathbf{c} = \sum_{i=1}^N \gamma_i^c \mathbf{n}_i^c, \qquad \gamma_i^c = \frac{\exp(r_i^c)}{\sum_{j=1}^N \exp(r_j^c)}, \tag{22}$$

where γ_i^u and γ_i^c denote the attention weight of user's i-th clicked news and candidate news respectively.

3.7 Relevance Measuring and Model Training

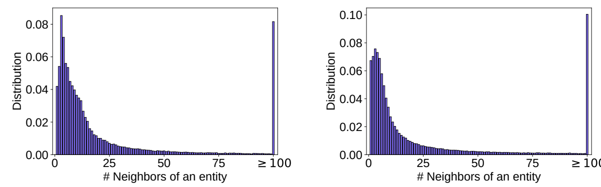
Following Okura et al. [22], we adopt dot product between candidate-aware user representation $\mathbf{u}$ and personalized candidate news representation $\mathbf{c}$ to measure the relevance $z \in \mathcal{R}$ between user's interest and candidate news content, i.e., $z = \mathbf{u}^T \cdot \mathbf{c}$. Then the candidate news is recommended based on the relevance score. Next, we introduce how we train the *KAIL* method. We utilize the negative sampling technique [9, 11] to construct the training dataset $\mathcal{S}$, where each positive sample is associated with a negative sample randomly selected from the same news impression. Then, we apply the BPR pairwise loss [24] to formulate the loss function of *KAIL*:

$$\mathcal{L} = -\frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|} \log(\sigma(z_+^i - z_-^i)), \tag{23}$$

where σ denotes the sigmoid function, z_+^i and z_-^i denote the relevance score of the i-th positive and negative sample respectively.

Table 1: Statistic information of the two datasets.

	MIND	*Feeds*
# News	73,897	1,126,508
# Users	50,000	50,605
# Impressions	320,775	210,000
# Clicks	489,350	473,697
Avg. # words in news title	11.78	11.90
Avg. # entities in news title	2.86	0.99



(a) The *MIND* dataset (b) The *Feeds* dataset

Figure 5: The number of neighbors of an entity on the knowledge graph.

4 EXPERIMENT

4.1 Datasets and Experimental Settings

To evaluate the effectiveness of the *KAIL* method, we conducted experiments based on a public dataset [38] (named *MIND*[4]) constructed by user logs collected from Microsoft News[5] and another dataset (named *Feeds*) constructed by user logs collected from a commercial news feeds platform.[6] User logs in *MIND* were sampled from October 12 to November 22, 2019 (six weeks), where logs in the last week were used for evaluation, logs in the penultimate week were used for model training, and others were used to construct users' historical clicked news. *MIND* also contained entities which were extracted from news and linked to WikiData[7] and corresponding entity embeddings trained based on the knowledge tuples extracted from WikiData via the TransE method [3]. User logs in *Feeds* were collected from January 23 to April 01, 2020 (thirteen weeks), where we randomly sampled 100,000 and 10,000 impressions from the first ten weeks to construct the training dataset and validation dataset independently, and randomly sampled 100,000 impressions from the last three weeks to construct the test dataset. Users' previous clicks before the time of impression were used to construct users' historical clicks used in the same impression. Following Wu et al. [38], we also extracted entities in news titles and pre-trained their embeddings based on the WikiData. The detailed information is summarized in the Table 1 and Fig. 5.

Next, we introduce our experimental settings. The word embedding vectors were initialized by the 300-dimensional glove embeddings [23]. The entity embedding vectors were initialized by the 100-dimensional TransE embeddings [3]. The knowledge graph

[4]We used the small version they released in https://msnews.github.io/index.html.
[5]https://www.msn.cn/en-us
[6]The platform is anonymized for double-blind review. Our data and code will be publicly available.
[7]https://www.wikidata.org/wiki/Wikidata:MainPage

Table 2: Performance of different methods on the two real-world datasets. *The improvement of our *KAIL* method over other baseline methods is significant at the level p < 0.01.

	MIND				*Feeds*			
	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
EBNR	63.86±0.21	29.70±0.15	32.06±0.23	37.69±0.15	63.44±0.39	27.97±0.25	32.01±0.32	37.57±0.35
DKN	65.75±0.18	31.14±0.12	33.71±0.13	39.38±0.16	62.91±0.26	28.08±0.20	32.20±0.24	37.75±0.22
DAN	65.74±0.38	30.84±0.21	33.42±0.21	39.12±0.25	62.65±0.49	27.79±0.32	31.79±0.40	37.37±0.39
NAML	65.84±0.18	31.26±0.23	33.86±0.26	39.52±0.20	64.24±0.38	28.81±0.21	33.06±0.28	38.52±0.29
NPA	65.71±0.28	31.13±0.26	33.79±0.27	39.48±0.25	63.69±0.75	28.51±0.47	32.74±0.64	38.27±0.62
LSTUR	66.12±0.24	31.56±0.20	34.11±0.25	39.78±0.20	64.66±0.33	29.04±0.26	33.44±0.32	38.82±0.30
NRMS	66.37±0.19	31.61±0.10	34.34±0.13	39.99±0.14	65.15±0.13	29.29±0.12	33.78±0.13	39.24±0.13
KRED	66.45±0.21	31.63±0.16	34.36±0.18	39.97±0.16	65.47±0.07	29.59±0.04	34.15±0.05	39.69±0.05
FIM	66.80±0.71	32.02±0.43	34.75±0.51	40.42±0.53	65.67±0.20	29.83±0.24	34.51±0.31	39.97±0.25
KAIL	**67.98**±0.13	**32.79**±0.18	**35.74**±0.24	**41.43**±0.19	**66.45**±0.13	**30.27**±0.09	**35.04**±0.09	**40.43**±0.12

used in experiments was WikiData. In addition, dimensions of output vectors of the CNN network and the transformer were set to 400. The output vectors of multi-head self-attention networks in the graph attention and co-attention networks were 100-dimensional. Besides, the dimensions of attention queries in the co-attention networks were set to 100. To alleviate the overfitting issue, we applied the dropout technique [26] to our method with 0.2 dropout probability. We utilized Adam optimizer [14] to optimize the trainable parameters of our method. Each experiment was repeated five times, and we report the average results and standard deviations.

4.2 Performance Evaluation

We compare *KAIL* with several state-of-the-art personalized news recommendation methods, which are listed as follow:

- *EBNR* [22]: learning news content representations from news bodies via an auto-encoder and learning user interest representations from users' clicked news via a GRU network [5].
- *DKN* [32]: using a knowledge-aware convolutional network to learn news representations from words and entities of news titles, and learning user representations from their historical clicks via a candidate-aware attention network.
- *DAN* [40]: learning news representations from words and entities of news titles, and using an attentive LSTM network [10] for user interest modeling.
- *NAML* [33]: applying an attentive multi-view learning framework to learn news representations from news titles, bodies, categories, and sub-categories.
- *NPA* [34]: using personalized attention networks to learn news representations from news titles and user representations from clicked news.
- *LSTUR* [1]: modeling user interest as the combination of short-term interest inferred from user's recent clicked news via a GRU network and long-term interest which is captured via user ID embeddings.
- *NRMS* [36]: using multi-head self-attention networks to learn news representations from news titles and user interest representations from users' reading history.
- *KRED* [19]: learning knowledge representations for news from the entities in news and their neighbors in knowledge graph via a graph attention network.
- *FIM* [31]: capturing user interest in candidate news based on the similarities between words in users' historical clicks and candidate news.

The experimental results are summarized in the Table 2, according to which we have several findings. First, *KAIL* significantly outperforms other baseline methods which learns news representations from news texts, such as *LSTUR* and *NRMS*. Besides, *KAIL* also significantly outperforms other baseline methods which learn news representations from texts and entities of news, such as *DKN* and *KRED*. This is because these methods independently learn representations for different news and cannot effectively capture the fine-grained relatedness between user's clicked news and candidate news when learning their representations. Different from these methods, *KAIL* learns knowledge-aware interactive representations for user's clicked news and candidate news from the interactions between their entities and interactions between their titles, which can effectively capture the fine-grained relatedness between them. Second, *KAIL* also outperforms *FIM* which captures fine-grained user interest in news content based on the similarities between words in user's clicked news and candidate news. This is because *FIM* ignores relatedness between entities of user's clicked news and candidate news which can provide rich information to understand their content [32]. Besides, relatedness between entities also should be exploited in a fine-grained manner since relatedness between entities usually has different importance for modeling the relatedness between different methods. Different from *FIM*, *KAIL* also learns interactive knowledge representations of user' clicked news and candidate news by capturing the fine-grained interactions between their entities to enhance their news representations, which can capture their fine-grained relatedness more effectively.

4.3 Ablation Study

In this section, we conduct two ablation studies to evaluate the effectiveness of *KAIL*. We first evaluate the effectiveness of different

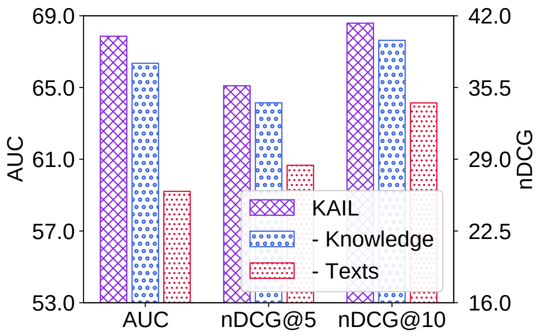


Figure 6: Performance of *KAIL* with different information for news content modeling.

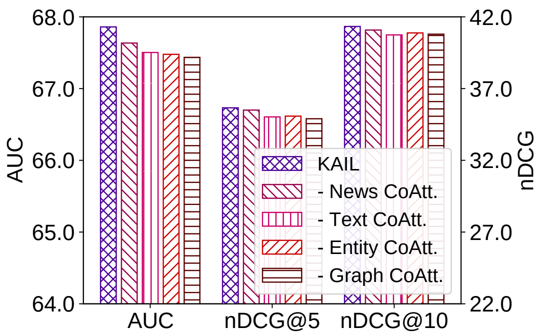


Figure 7: Ablation study on different components of *KAIL*, where CoAtt. denotes the co-attention network.

information, i.e., texts and knowledge, for news content modeling. The experimental results are shown in Fig. 6, from which we have several observations.[8] First, removing texts seriously hurts the performance of *KAIL*. This is because texts usually contain rich information on news content and are vitally important for news content understanding [38]. Removing texts makes the news representations lose much important information and cannot accurately the model news content. Second, removing knowledge in news content modeling also makes the performance of *KAIL* decline significantly. This is because textual information is usually insufficient to understand news content [19, 32]. Fortunately, knowledge graph contains rich relatedness between different entities. Moreover, relatedness between entities in user's clicked news and candidate news can provide rich information beyond texts for modeling their relatedness. Thus, incorporating knowledge into personalized news recommendation has the potential to improve the accuracy of recommendation.

Next, we evaluate the effectiveness of different components of *KAIL*. The Fig. 7 shows the experimental results, from which we have several findings. First, removing the news co-attention network in *interactive user-candidate encoder* makes the performance of *KAIL* decline. This is because user interest may be diverse, and

[8]In the following section, we only show the experimental results on the *MIND* dataset.

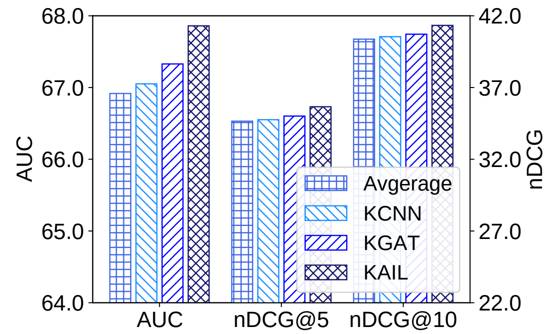


Figure 8: Performance of *KAIL* and its variations with different knowledge modeling methods.

only part of user's clicked news are informative for modeling the relevance between user interest and candidate news. Besides, candidate news content may contain multiple aspects and different users may be interested in different aspects. Thus, learning candidate-aware user interest and personalized candidate news representation via a news co-attention network can better capture user interest in candidate news. Second, removing the text co-attention network also hurts the performance of *KAIL*. This is because news content may have multiple aspects, and an aspect of news usually has different importance for modeling its relatedness with different news. Thus, the text co-attention network which learns interactive text representations for user's clicked news and candidate news from the fine-grained interactions between their titles can better capture their fine-grained relatedness. Third, removing both the graph co-attention network and entity co-attention network make the performance of *IKSN* decline. This is because an entity usually has rich relatedness with different entities and relatedness between entities usually has different importance for modeling the relatedness between different news. Both the graph co-attention network which learns entity representations by selecting their informative neighbors and the entity co-attention network which learns knowledge representations from interactions between their entities in news makes *KAIL* capture the fine-grained relatedness between user's clicked news and candidate news more effectively.

4.4 Effectiveness of the Knowledge Encoder

In this section, we evaluate the effectiveness of the *interactive knowledge encoder* in *KAIL* by comparing *KAIL* with its variations which model knowledge of user' clicked and candidate news without being aware of each other. The first one (named *Average*) averages the embeddings of entities in news and their neighbors within K hops as the knowledge representations. The second one (named *KCNN*) learns knowledge representations from entities and their neighbors via the knowledge-aware convolutional network proposed in *DKN* [32]. The third one (named *KGAT*) uses the knowledge modeling method of *KRED* that using a graph attention network to learn knowledge representations from entities in news and their neighbors on the knowledge graph. Besides, all of these variations

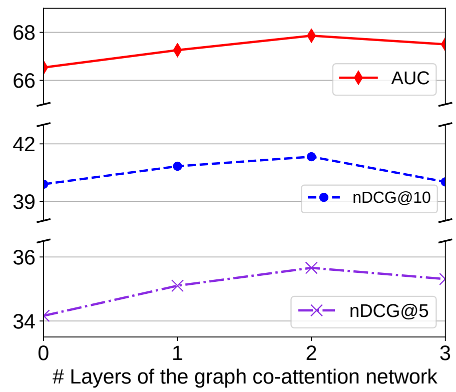


Figure 9: Model performance under different number of layers of the graph co-attention network, i.e., K.

have the same text modeling method with *KAIL* for fair comparisons. The experimental results are shown in Fig. 8, from which we have several findings.

First, *Average* has the worst performance among these methods. This is because different entities in news and their neighbors usually have different informativeness for news content understanding. Since *Average* ignores the relative importance of different entities, it cannot effectively model news content based on entities. Second, *KGAT* outperforms *KCNN*. This is because there is usually relatedness between different neighbors of an entity. *KCNN* only uses the average embeddings of neighbors of entities in news to enhance their representations and ignores such relatedness. Different from *DKN*, *KGAT* utilizes a graph attention network to model the relatedness between neighbor entities, which can learn more accurate entity representations. Third, *KAIL* outperforms both *KCNN* and *KGAT* significantly. This is because *KCNN* and *KGAT* model knowledge for user's clicked news and candidate news without being aware of each other. However, as shown in Fig. 5, an entity on the knowledge graph usually has rich relatedness with many different entities and the informative relatedness for modeling relatedness between different news are usually different. Thus, it is difficult for these methods to effectively capture fine-grained relatedness between user's clicked news and candidate news from the fine-grained interactions between their entities. Different from them, *KAIL* uses an *interactive knowledge encoder* to learn knowledge representations for user's clicked news and candidate news by capturing the relatedness between their entities, which can effectively capture the fine-grained relatedness between them.

4.5 Influence of the Number of Layers of GCAT

In this section, we evaluate the influence of the number of layers of the graph co-attention network, i.e., K, on the performance of *KAIL*. We show the performance of *KAIL* under different value of K, i.e., $\{0, 1, 2, 3\}$ in Fig. 9, from which we have two observations. First, the performance of the *KAIL* method first increases with the increase of K. This is because the relatedness between entities is informative for understanding the relatedness between news. Besides, the GCAT network stacked for K layers can learn representations for entities in news from their neighbors within K hops. When K is too small, the relatedness between user's clicked news and candidate news cannot be fully explored based on their entities, which is harmful to the recommendation accuracy. Second, when K is too large, the performance of the *KAIL* method begins to decline. This is because when K becomes too large, too many multi-hop neighbors are incorporated for modeling the relatedness between user's clicked news and candidate news, which may bring much noise to the model and hurt the recommendation accuracy. Thus, a moderate value of K, i.e., 2, is suitable for our method.

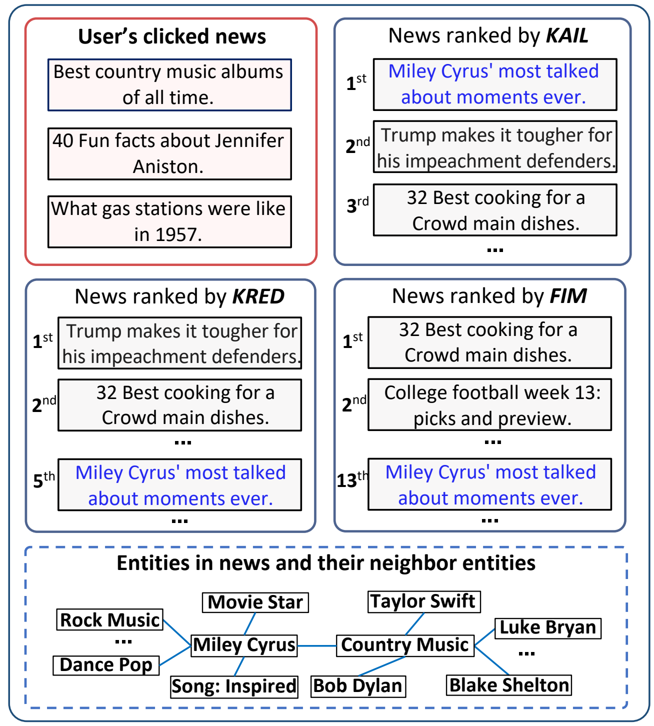


Figure 10: An illustrative case of personalized news recommendation. Only the news in blue is clicked by the user in the same impression.

4.6 Case Study

In this section, we conduct a case study to show the effectiveness of *KAIL* by comparing it with *FIM* and *KRED* since the former achieves the best performance among baseline methods without knowledge and the latter achieves the best performance among knowledge-aware baseline methods in the Table 2. In Fig. 10, we show the reading history of a randomly sampled user, and the news recommended by these methods in the same impression where the user only clicked one candidate news. According to Fig. 10, we have several observations. First, both *KRED* and *KAIL* rank the candidate news clicked by the user higher than *FIM*. This is because it is difficult to understand the relatedness between user's clicked news and the candidate news based on their textual information. However, according to the knowledge graph, we can find that the entity "Country Music" in the first historical clicks of the user has a relation with the entity "Miley Cryus" in the candidate news clicked by the user since Miley Cryus is a representative singer of country music. Thus, based the information provided by the knowledge graph, *KRED* and *KAIL* can have a better understanding of the content of these two news. Second, *KAIL* ranks the candidate news clicked by the user higher than *KRED*. This is because both of these two entities have rich relatedness with many other neighbor entities on the knowledge graph. For example, besides "Miley Cyrus", the entity "Country Music" has relatedness with many other representative singers such as Bob Dylan, Talyor Swift, and so on. In addition, the entity "Miley Cryus" also has relatedness with the entities of other areas which Miley Cryus is skilled in, such as rock music, dance-pop and so on. Thus, it is difficult for *KRED* to effectively capture the fine-grained relatedness between the two news from their entities since *KRED* independently learns the knowledge-aware representations for user's clicked news and candidate news. Different from *KRED*, *KAIL* learns knowledge-aware interactive representations of user's clicked news and candidate news from the fine-grained interactions between their entities and fine-grained interactions between their titles, which can effectively capture the fine-grained relatedness between them. Thus, *KAIL* can better capture user interest in candidate news than *KRED*.

5 CONCLUSION

In this paper, we propose a personalized news recommendation framework with knowledge-aware interactive learning (named *KAIL*), which can effectively model the fine-grained relatedness between users' clicked news and candidate news. The core module of *KAIL* is a *knowledge-aware interactive news encoder*. In detail, it contains an *interactive knowledge encoder* to capture the fine-grained interactions between entities of user's clicked news and candidate news. In the *interactive knowledge encoder*, we first propose a graph co-attention network to learn representations for entities by selecting their neighbors which are informative for modeling the relatedness between their entities. Then we propose to use an entity co-attention network to learn interactive knowledge representations for user's clicked news and candidate news by capturing the interactions between their entities. Besides, it also contains an *interactive text encoder* to learn interactive text representations of user's clicked news and candidate news by capturing the fine-grained interactions between their contexts. The news representation is the aggregation of the text representation and knowledge representation of the same news. Moreover, we propose an *interactive user-candidate encoder* to learn candidate-aware user representation and personalized candidate news representation to better capture the relevance between user interest and candidate news. We conduct extensive experiments on two real-world datasets. The experimental results show that our *KAIL* method can significantly outperform other baseline methods.

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*. 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*. 195–202.
[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*. 2787–2795.
[4] Qianfeng Chu, Gongshen Liu, Huanrong Sun, and Cheng Zhou. 2019. Next news recommendation via knowledge-aware sequential model. In *CCL*. 221–232.
[5] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Workshop on Deep Learning*.
[6] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *WWW*. 271–280.
[7] Jie Gao, Xin Xin, Junshuai Liu, Rui Wang, Jing Lu, Biao Li, Xin Fan, and Ping Guo. 2018. Fine-grained deep knowledge-aware network for news recommendation with self-attention. In *WI*. 81–88.
[8] Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. 2017. Knowledge transfer for out-of-knowledge-base entities: a graph neural network approach. In *IJCAI*. 1802–1808.
[9] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *SIGIR*. 355–364.
[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* (1997), 1735–1780.
[11] Zhengshen Jiang, Hongzhi Liu, Bin Fu, Zhonghai Wu, and Tao Zhang. 2018. Recommendation in heterogeneous information networks based on generalized random walk model and bayesian personalized ranking. In *WSDM*. 288–296.
[12] 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*. 1855–1858.
[13] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*. 1746–1751.
[14] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.
[15] Michal Kompan and Mária Bieliková. 2010. Content-based news recommendation. In *International conference on electronic commerce and web technologies*. 61–72.
[16] 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* (1997), 77–87.
[17] 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.
[18] Chen Lin, Runquan Xie, Xinjun Guan, Lei Li, and Tao Li. 2014. Personalized news recommendation via implicit social experts. *Information Sciences* (2014), 1–18.
[19] Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. 2020. KRED: Knowledge-aware document representation for news recommendations. In *RecSys*. 200–209.
[20] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *IUI*. 31–40.
[21] Zheng Liu, Yu Xing, Fangzhao Wu, Mingxiao An, and Xing Xie. 2019. Hi-Fi ark: deep user representation via high-fidelity archive network. In *IJCAI*. 3059–3065.
[22] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *KDD*. 1933–1942.
[23] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.
[24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.
[25] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 395–405.
[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR* (2014), 1929–1958.
[27] 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*. 6000–6010.
[28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.
[29] Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge.. In *AAAI*. 855–860.
[30] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*. 448–456.
[31] Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *ACL*. 836–845.
[32] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *WWW*. 1835–1844.
[33] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with attentive multi-view learning. *IJCAI* (2019), 3863–3869.

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

[35] 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*. 4876–4885.

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

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

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

[39] 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.

[40] Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Guo Li. 2019. DAN: Deep attention neural network for news recommendation. In *AAAI*. 5973–5980.