Personalized News Recommendation with Knowledge-aware Interactive Matching

Tao Qi[1], Fangzhao Wu[2], Chuhan Wu[1], Yongfeng Huang[1]
[1]Department of Electronic Engineering & BNRist, Tsinghua University, Beijing 100084
[2]Microsoft Research Asia, Beijing 100080, China
{taoqi.qt,wufangzhao,wuchuhan15}@gmail.com,yfhuang@tsinghua.edu.cn

ABSTRACT

The core of personalized news recommendation is accurate matching between candidate news and user interest. Most existing news recommendation methods usually model candidate news from its textual content and model users' interest from their clicked news, independently. However, a news article may cover multiple aspects and entities, and a user may have multiple interests. Independent modeling of candidate news and user interest may lead to inferior matching between news and users. In this paper, we propose a knowledge-aware interactive matching framework for personalized news recommendation. Our method can interactively model candidate news and user interest to learn user-aware candidate news representation and candidate news-aware user interest representation, which can facilitate the accurate matching between user interest and candidate news. More specifically, we propose a knowledge co-encoder to interactively learn knowledge-based news representations for both clicked news and candidate news by capturing their relatedness in entities with the help of knowledge graphs. In addition, we propose a text co-encoder to interactively learn text-based news representation for clicked news and candidate news by modeling the semantic relatedness between their texts. Besides, we propose a user-news co-encoder to learn candidate news-aware user interest representation and user-aware candidate news representation from the knowledge- and text-based representations of candidate news and clicked news for better interest matching. Through extensive experiments on two real-world datasets, we demonstrate our method can effectively improve the performance of news recommendation.

CCS CONCEPTS

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

KEYWORDS

News Recommendation, User Interest, Interactive Matching

ACM Reference Format:
Tao Qi[1], Fangzhao Wu[2], Chuhan Wu[1], Yongfeng Huang[1]. 2021. Personalized News Recommendation with Knowledge-aware Interactive Matching. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021).* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Online news platforms such as Yahoo! News and Bing News, have attracted a huge number of users to consume news information [22, 34]. However, since massive new published news articles are collected by these platforms every day, users often have difficulties in finding the news information they need [35, 47]. 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 [1, 33]. Thus, the study on personalized news recommendation has attracted much attention from both academia and industry [1, 2, 11, 14, 20, 43, 46].

Accurate matching between user interest and candidate news is critical for personalized news recommendation [32, 33]. Existing methods usually model candidate news from its textual information, and infer user interest from user's click history, independently [22, 38]. For example, Wu et al. [35] independently applied a word-level attention network with a personalized query to learn representations of candidate news its title and applied a news-level personalized attention network to learn user interest representation from clicked news. They further performed the inner product of user interest representation and candidate news representation for interest matching. However, a candidate news article may contain multiple aspects and entities [19, 34], and a user may have multiple interests [33]. Thus, independent modeling of candidate news and user interest may be inferior for the interest matching [32].

Our paper is motivated by the following observations. First, a candidate news may cover different aspects and entities, and a user may have multiple interests. For example, the candidate news in Fig. 1 is related to a movie and a technical corporation, and covers several entities, i.e., "Movie Cats" and "Netflix". Besides, the example user in Fig. 1 is interested in different areas such as politics, sports, and entertainment. In addition, we can find the candidate news can only match a specific user interest, i.e., entertainment, and user interest can only match a specific candidate news aspect, i.e., movie. Thus, it is inferior for matching user interest with candidate news if they are independently modeled. Second, the matching between user interest and candidate news is usually implied in the matching between texts of clicked news and candidate news. For instance, we can infer the user may be interested in candidate news from the semantic relatedness between text "trending song" in clicked news and "popular movie" in candidate news since both of them are popular work. Third, with the help of the knowledge graph, the matching between entities in clicked news and candidate news is

SIGIR 2021, July 2021, Canada

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

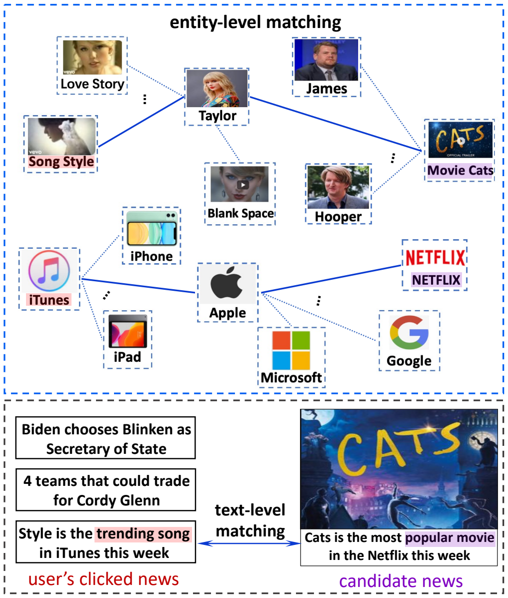
entity-level matching
Love Story
Taylor
James
Song Style
Movie Cats
Blank Space
Hooper
iPhone
NETFLIX
iTunes
Apple
iPad
Microsoft
Google
Biden chooses Blinken as Secretary of State
4 teams that could trade for Cordy Glenn
Style is the trending song in iTunes this week
text-level matching
Cats is the most popular movie in the Netflix this week
user's clicked news
candidate news

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

also informative for understanding user interest in candidate news. For example, we can find the entity "Song Style" in clicked news has inherent 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, from which we can infer the user may have interest in the candidate news. Thus, exploiting the relatedness between clicked news and candidate news at both text and entity levels effectively is beneficial for interest matching.

In this paper, we propose a knowledge-aware interactive matching framework for personalized news recommendation (named *KIM*). Our method can interactively model candidate news and user interest to learn candidate news-aware user interest representation and user-aware candidate news representation to match user interest and candidate news more accurately. In the framework, we propose a knowledge co-encoder to model user interest in candidate news from the relatedness between entities in clicked news and candidate news with the help of knowledge graphs. More specifically, we first propose a graph co-attention network to learn representations of entities from the knowledge graph by selecting and aggregating their neighbors which are informative for interest matching. We further propose to use an entity co-attention network to interactively learn knowledge-based representations of both clicked news and candidate news by capturing relatedness between their entities. Moreover, we also propose a text co-encoder to interactively learn text-based representations for user's clicked news and candidate news by modeling semantic relatedness between their texts. The unified representation of news is formulated as the aggregation of its knowledge- and text-based representation. In addition, we further propose a user-candidate co-encoder to build candidate news-aware user interest representation and user-aware candidate news representation from representations of clicked news and candidate news to better model user interest candidate news. Finally, the candidate news is ranked based on the relevance between representations of candidate news and user interest. We conduct extensive experiments on two real-world datasets and show that our method can effectively improve the performance of news recommendation and outperform other baseline methods.

2 RELATED WORK

Personalized news recommendation is an important task for online news services [5, 18] and has been widely studied in recent years [15, 17, 25, 31, 36, 37, 42, 44]. Existing methods usually model candidate news from its content and model user interest from clicked news independently, and then recommend candidate news based on its matching with user interest [6, 33, 34, 38, 39]. For example, Okura et al. [22] represented candidate news from its bodies via a de-noising auto-encoder and represented user interest from user's click history via a GRU network, independently. They further performed dot product between representations of user interest and candidate news to measure their relevance. Wu et al. [38] adopted a multi-head self-attention network for modeling candidate news from its title and another multi-head self-attention network for modeling user interest from user's click history. Liu et al. [19] proposed to learn knowledge-based candidate news representation from entities in news title and their neighbors on the knowledge graph and learn user interest representation from user's clicked news via an attention network. In addition, these methods also used dot product to model the relevance between user interest and candidate news. In general, a candidate news may cover multiple aspects and entities [19, 34], and a user may have multiple interests [33]. Only a part of candidate new aspects and user interests are usually useful for matching user interest with candidate news. However, these methods model candidate news and user interest independently, which may be inferior for the further interest matching. Different from these methods, in *KIM* we propose a knowledge-aware interactive matching framework to interactively model candidate news and user interest with the consideration of their relatedness, which can better match user interest with candidate news.

Some methods model user interest in a candidate-aware way [33, 47]. For example, Wang et al. [33] proposed to learn news representations from embeddings of aligned words and entities in news titles via a multi-channel CNN network and applied a candidate-aware attention network to learn user interest representation by aggregating representations of clicked news based on their relevance with candidate news. They further used a dense network to model the relevance of user interest and candidate news. Zhu et al. [47] proposed to learn news representations from words and entities in news titles via multiple CNN networks and learn user interest representations from historical clicks via a LSTM network and an candidate-aware attention network. They adopted cosine similarity of user interest and candidate news representation to model their relevance. In fact, candidate news may contain multiple aspects and entities [19, 34] and only a part of them may match user interest. However, these methods model candidate news without the consideration of the target user, which maybe inferior for further matching user interest with candidate news. Different from these methods, our *KIM* method models candidate news with the consideration of target user. In addition, these methods model clicked

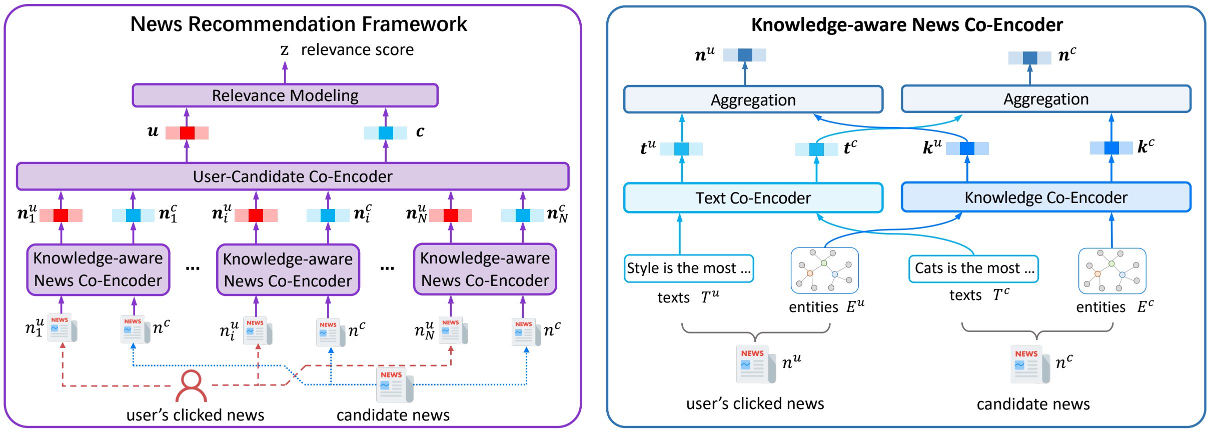

Figure 2: The news recommendation framework and knowledge-aware news co-encoder of the *KIM* method.

news and candidate news without the consideration of their relatedness, which may also be suboptimal for further measuring relevance between candidate news and user interest inferred from clicked news. Different from these methods, *KIM* can interactively learn representations of both clicked news and candidate news for better interest matching.

3 METHODOLOGY

We first introduce the problem definition of personalized news recommendation. Next, we introduce our knowledge-aware interactive matching framework for personalized news recommendation (named *KIM*).

3.1 Problem Formulation

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 texts T of its texts and entities E in its texts. Besides, there is a knowledge graph $\mathcal{G}$ used to provide the relatedness between entities. It contains entities and relations between entities. Each entity e in $\mathcal{G}$ is associated with its embeddings $\mathbf{e}$ pre-trained based on the knowledge graph. In our method, we only utilize the links between entities to represent their relatedness and do not utilize the specific relations (e.g., located_at).

3.2 Framework of KIM

In this section, we introduce the news recommendation framework of *KIM*, which can interactively model candidate news and user interest for better interest matching. As illustrated in Fig. 2, *KIM* contains two major modules. The first one is a *knowledge-aware news co-encoder*, which interactively learns the knowledge-aware representations of a user's clicked news and the candidate news by capturing their relatedness at both text and entity levels. The second one is a *user-candidate co-encoder*, which interactively learns candidate news-aware user interest representation $\mathbf{u}$ and user-aware candidate news representation $\mathbf{c}$ from the representations of user's clicked news and candidate news generated by the *knowledge-aware news co-encoder*. Finally, we match candidate news with user interest based on the relevance between the candidate news-aware user interest representation and user-aware candidate news representation. Next, we introduce each module in detail.

3.3 Knowledge-aware News Co-Encoder

In this section, we introduce the framework of the *knowledge-aware news co-encoder*, which interactively learns representations of a user's clicked news n^u and candidate news n^c from texts and entities of their texts. As shown in Fig. 2, it contains three sub-modules. The first one is a *knowledge co-encoder* (denoted as Φ_k), which interactively learns knowledge-based representations $\mathbf{k}^u \in \mathcal{R}^{d_k}$ and $\mathbf{k}^c \in \mathcal{R}^{d_k}$ for clicked news n^u and candidate news n^c from the relatedness 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-based news representation dimensions, E^u and E^c denote entities in news n^u and n^c respectively. The second one is a *text co-encoder* (denoted as Φ_t), which interactively learns text-based 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 to model user interests in candidate news from the semantic relatedness between their texts:

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

where d_t denotes text-based news representation dimensions, T^u and T^c denote texts of news n^u and n^c respectively. Finally, we project the knowledge- and text-based 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 representation of user's clicked news n^u, $\mathbf{n}^c \in \mathcal{R}^{d_n}$ denotes the corresponding knowledge-aware representation of candidate news n^c, d_n denotes news representation dimensions, $[\cdot;\cdot]$ denotes the concatenation operation, and $\mathbf{P}_n \in \mathcal{R}^{d_n \times (d_t + d_k)}$ is the trainable projection matrix.

3.3.1 Knowledge Co-Encoder. We introduce the proposed *knowledge co-encoder*, which interactively learns the knowledge-based

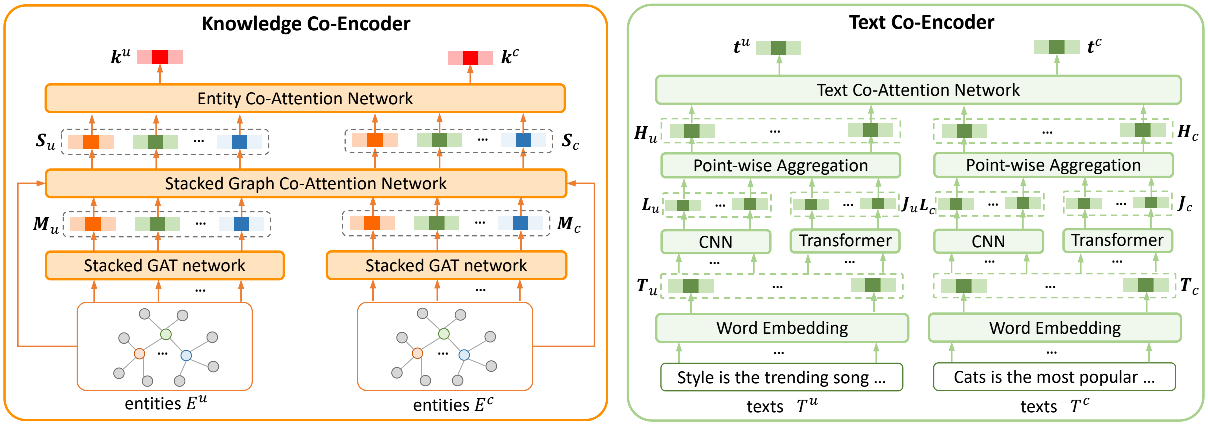

Figure 3: The architecture of the knowledge co-encoder and text co-encoder.

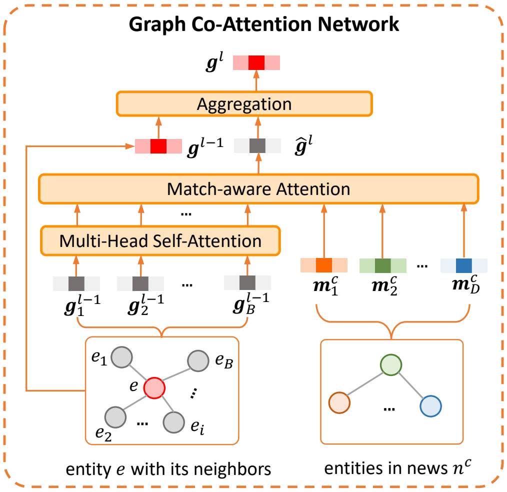

Figure 4: The architecture of the GCAT network.

representations of user's clicked news n^u and candidate news n^c. It aims to better represent these news for interest matching from relatedness between entities E^u and E^n in user's clicked news and candidate news with the help of the knowledge graph $\mathcal{G}$. As shown in Fig. 3, it contains three components. To summarize the information for each entity in E^u or E^c from their neighbors within K hops, we first utilize a graph attention (GAT) network [29] 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 is the 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 [7, 30]. Besides, relatedness among entities usually provides different informativeness to model the relatedness between clicked news and candidate news for interest matching. For example, Fig. 1 shows the entity "Movie Cats" has many neighbor entities, such as its director "James", chief actor "Hooper", chief actress "Taylor" and so on. Only the entity "Taylor" is informative for modeling the relatedness between clicked news and candidate news since it is also the singer of the entity "Song Style" in clicked news. To better select informative relatedness between entities for matching candidate news with user interest, 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 representations of its neighbors guided by entities in news n^c. More specifically, we first apply a multi-head self-attention network [28] to the representations of its neighbor entities generated by the $(l-1)$-th GCAT network[1] to model the conceptual relatedness between different neighbor entities. Next, we propose a match-aware attention network to aggregate neighbor entities of entity e based on their relevance with entities in news n^c 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 as:

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

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 v_i^u is the i-th element of vector $\mathbf{v}^u$ and 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: $\mathbf{g}^l = \mathbf{P}_e[\hat{\mathbf{g}}^l; \mathbf{g}^{l-1}]$, where $\mathbf{P}_e \in \mathcal{R}^{d_k \times 2d_k}$ is the projection

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

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, where $\mathbf{s}_i^u$ is the representation of the $i-$th entity in clicked news n^u. 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 from relatedness between their neighbors and entities in clicked news, where $\mathbf{s}_i^c$ is the representation of the $i-$th entity in candidate news n^c.

The third one is an entity co-attention network. Entities in clicked news and candidate news usually have different informativeness for interest matching. For example, according to Fig. 1, in clicked news, the entity "Song Style" is more informative than the entity "iTunes" for matching user interest with candidate news since the entity "Song Style" has inherent relatedness with the entity "Movie Cats" in candidate news. Thus, we apply an entity co-attention network to interactively learn knowledge-based representations for news n^u and n^c by capturing relatedness between their entities. In detail, we first calculate an affinity matrix $\mathbf{C}_e \in \mathcal{R}^{D \times D}$ to measure the relevance among entities in news n^u and n^c:

$$\mathbf{C}_e = \mathbf{S}_c^T \mathbf{W}_c^k \mathbf{S}_u, \tag{7}$$

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

$$\mathbf{a}^u = \mathbf{q}_k^T \cdot \tanh(\mathbf{W}_s^k \mathbf{S}_u + \mathbf{W}_h^k \mathbf{S}_c f(\mathbf{C}_e)), \tag{8}$$

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

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 we obtain knowledge-based representations $\mathbf{k}^u \in \mathcal{R}^{d_k}$ and $\mathbf{k}^c \in \mathcal{R}^{d_k}$ of clicked news and candidate news by aggregating their entities respectively:

$$\mathbf{k}^u = \sum_{i=1}^{D} \alpha_i^u \mathbf{s}_i^u, \qquad \alpha_i^u = \frac{\exp(a_i^u)}{\sum_{j=1}^{D} \exp(a_j^u)}, \tag{10}$$

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

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.

3.3.2 Text Co-Encoder. As shown in Fig. 3, *text co-encoder* interactively learns the text-based representations for user's clicked news n^u and candidate news n^c. It aims to better model user interests in candidate news from relatedness between texts of their texts (T^u and T^c). We first independently learn contextual representations for words in texts T^u and T^c. More specifically, take texts T^u as an example, we first convert it into an embedding vector sequence $\mathbf{T}_u \in \mathcal{R}^{d_g \times M}$ via a word embedding layer, where d_g denotes word embedding dimensions, and M denotes the number of words in texts. Next, since both local and global contexts are important for text modeling [34, 38], we apply a CNN network [12] and a transformer network [28] to $\mathbf{T}_u$ to learn both local- and global-contextual word representations respectively, i.e., $\mathbf{L}_u \in \mathcal{R}^{d_t \times M}$ and $\mathbf{J}_u \in \mathcal{R}^{d_t \times M}$. Then, we add the local- and global-contextual representations of each word and obtain their unified representations $\mathbf{H}_u = \{\mathbf{h}_i^u\}_{i=1}^M \in \mathcal{R}^{d_t \times M}$, where $\mathbf{h}_i^u \in \mathcal{R}^{d_t}$ is the representation of the i-th word in texts 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 texts T^c in the same way, where $\mathbf{h}_i^c \in \mathcal{R}^{d_t}$ is the i-th word representation in texts T^c.

Finally, note that different semantic aspects in clicked news and candidate usually have different importance for matching user interest with candidate news [40]. For example, given a clicked news "Apple's plans to make over-ear headphones.", it contains two semantic aspects, i.e., "Apple's product plan" and "headphones". The former is important for matching user interest with candidate news "The best headphones of 2020." since users interested in headphones may click both of them. While the latter is important for matching user interest with candidate news "iPhone 12 cases buyer's guide." since users interested in the product of Apple may read them. Thus, we apply a text co-attention network [26, 41] to interactively learn text-based representations of news n^u and n^c by capturing semantic relatedness between their texts for interest matching . Specifically, we first calculate the affinity matrix $\mathbf{C}_t \in \mathcal{R}^{M \times M}$ measuring the semantic relevance between different words in texts T^u and T^c:

$$\mathbf{C}_t = \mathbf{H}_c^T \mathbf{W}_c^t \mathbf{H}_u, \tag{12}$$

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 $\mathbf{C}_t$:

$$\mathbf{b}^u = \mathbf{q}_t^T \cdot \tanh(\mathbf{W}_s^t \mathbf{H}_u + \mathbf{W}_h^t \mathbf{H}_c f(\mathbf{C}_t)), \tag{13}$$

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

where $\mathbf{q}_t \in \mathcal{R}^{d_q}$ is 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 trainable parameters. Finally, we learn text-based 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:

$$\mathbf{t}^u = \sum_{i=1}^{M} \beta_i^u \mathbf{h}_i^u, \qquad \beta_i^u = \frac{\exp(b_i^u)}{\sum_{j=1}^{M} \exp(b_j^u)}, \tag{15}$$

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

where β_i^u and β_i^c is weight of the i-th word in texts T^u and T^c.

3.4 User-Candidate Co-Encoder

We introduce our proposed *user-candidate co-encoder*, which learns candidate news-aware user interest representation and user-aware candidate news representation from representations of user's clicked news and candidate news. Usually, interests of a user is diverse, and only part of them can be matched with a candidate news [21?]. Thus learning candidate news-aware user interest representation can better model user interest for matching candidate news. Similarly, a candidate news may cover multiple aspects, and a user may only be interested in part of them [34, 35]. Thus learning user-aware candidate news representation is also beneficial for interest matching. Thus, we apply a news co-attention network to learn candidate news-aware user representation and user-aware candidate news representation. More specifically, we first calculate the affinity matrix $\mathbf{C}_n \in \mathcal{R}^{N \times N}$ based on the 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:

$$\mathbf{C}_n = \mathbf{N}_c^T \mathbf{W}_c^n \mathbf{N}_u, \tag{17}$$

where N denotes the number of clicked news, $\mathbf{n}^u_i \in \mathcal{R}^{d_n}$ denotes the representation of user's i-th clicked news, $\mathbf{n}^c_i \in \mathcal{R}^{d_n}$ denotes the corresponding representation of candidate news, and $\mathbf{W}^n_c \in \mathcal{R}^{d_n \times d_n}$ 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:

$$\mathbf{r}^u = \mathbf{q}_n^T \cdot \tanh(\mathbf{W}^n_s \mathbf{N}_u + \mathbf{W}^n_h \mathbf{N}_c f(\mathbf{C}_n)), \tag{18}$$

$$\mathbf{r}^c = \mathbf{q}_n^T \cdot \tanh(\mathbf{W}^n_s \mathbf{N}_c + \mathbf{W}^n_h \mathbf{N}_u f(\mathbf{C}_n^T)), \tag{19}$$

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

$$\mathbf{u} = \sum_{i=1}^{N} \gamma^u_i \mathbf{n}^u_i, \qquad \gamma^u_i = \frac{\exp(r^u_i)}{\sum_{j=1}^{N} \exp(r^u_j)}, \tag{20}$$

$$\mathbf{c} = \sum_{i=1}^{N} \gamma^c_i \mathbf{n}^c_i, \qquad \gamma^c_i = \frac{\exp(r^c_i)}{\sum_{j=1}^{N} \exp(r^c_j)}, \tag{21}$$

where γ^u_i and γ^c_i denote attention weight of $\mathbf{n}^u_i$ and $\mathbf{n}^c_i$ respectively.

3.5 Relevance Modeling and Model Training

Following Okura et al. [22], we adopt dot product of candidate news-aware user representation $\mathbf{u}$ and user-aware candidate news representation $\mathbf{c}$ to measure the relevance $z \in \mathcal{R}$ of user interest and candidate news content, i.e., $z = \mathbf{u}^T \cdot \mathbf{c}$. Candidate news are further recommended to the user based on their relevance scores.

Next, we introduce how we train the *KIM* method. We utilize the negative sampling technique [8, 10] to construct the training dataset $\mathcal{S}$, where each positive sample is associated with U negative sample randomly selected from the same news impression. Then, we apply the NCE loss [23] to formulate the loss function:

$$\mathcal{L} = -\frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \log\left(\frac{\exp(z^i_+)}{\exp(z^i_+) + \sum_{j=1}^{U} \exp(z^i_j)}\right), \tag{22}$$

where σ denotes the sigmoid function, z^i_+ denotes the relevance score of the i-th positive sample, and z^i_j denotes the relevance score of the j-th negative sample selected for the i-th positive sample.

Finally, we briefly discuss the computational complexity of *KIM*. Different from the methods that model user and candidate news independently, *KIM* calculates representations of clicked news and candidate news collaboratively, which requires more computation resources because these representations cannot be prepared in advance. Fortunately, in practice we can calculate contextual word embeddings $\mathbf{H}$ and entity embeddings $\mathbf{M}$ of different news offline and cache them to save the computational cost.

4 EXPERIMENT

4.1 Datasets and Experimental Settings

In this section, we evaluate the performance of different methods based on a public dataset [45] (named *MIND*[2]) and another dataset (named *BING*) constructed by user logs collected from Bing news feeds.[3] *MIND* was constructed by six-week user logs sampled from Microsoft News during Oct. 12 to Nov. 22, 2019, where the training and validation set were constructed by user logs in the fifth week, and the test set was constructed by user logs in the sixth week. In *MIND* dataset, entities in news titles were extracted and linked to WikiData automatically. Their embeddings were trained based on the knowledge tuples extracted from WikiData via the TransE method [3]. The *BING* dataset was constructed by thirteen-week user logs during Jan. 23 to Apr. 01, 2020, where the training and validation set were constructed by 100,000 and 10,000 impressions randomly sampled from the first ten weeks respectively, and the test set was constructed by 100,000 impressions randomly sampled from the last three weeks. Following Wu et al. [45], in *BING* dataset we also extracted entities in news titles and pre-trained their embeddings based on the WikiData. In these two datasets, we used news titles as news texts, and only used entities in news titles. Besides, we used WikiData as the knowledge graph in experiments. It contained 3,275,149 entities, 20 types of relations and 29,824,585 links between entities. More detailed statistics is listed in Table 1.

Table 1: Detailed statistics of the *MIND* and *BING*.

	MIND	*BING*
# Users	50,000	50,605
# Impressions	320,775	210,000
# Clicks	489,350	473,697
# News	73,897	1,126,508
Avg. # words in news title	11.78	11.90
Avg. # entities in news title	2.86	0.99
Avg. # neighbors in KG	18.21	18.09

Next, we introduce all hyper-parameters of *KIM* and experiment settings. For each news, we only used the first 30 words and 5 entities in news titles. We randomly sampled 10 neighbors for each entity from the knowledge graph. Besides, we only used the recent 50 clicked news of each user. The word and entity embedding vectors were initialized by 300-dimensional glove embeddings [24] and 100-dimensional TransE embeddings [3], respectively. Due to limitation of GPU memory, we only fine-tuned word embeddings and did not fine-tune entity embeddings in experiments. In *text co-encoder*, the transformer contained 10 attention heads and output vectors of each head were 40-dimensional. Besides, the CNN network contained 400 filters. In *knowledge co-encoder*, all multi-head self-attention networks in the graph attention and co-attention networks contained 5 attention heads, and all of these heads output 20-dimensional vectors. Besides, all attention queries in *KIM* were set to 100-dimension. For effective model training we applied the dropout technique [27] with 0.2 dropout probability. We sampled 4 negative samples for each positive sample. We utilized Adam optimizer [13] to train *KIM* for 6 epochs with 5×10^{-5} learning rate. All hyper-parameters of *KIM* and other baseline methods were selected based on the validation dataset. We will release our codes implemented for these algorithms. Following previous works [34], we evaluated performance of different methods based on four ranking metrics, i.e., AUC, MRR, nDCG5, and nDCG10.

[2] We used the small version for quick experiments.

[3] Codes and data are released in https://github.com/JulySinceAndrew/KIM-SIGIR-2021.

Table 2: Performance of different methods on the two real-world datasets. *We perform a t-test on these results and *KIM* method significantly (at the level p < 0.01) outperforms all baseline methods.

	MIND				*BING*			
	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
EBNR	61.28±0.27	27.77±0.21	30.10±0.28	36.75±0.24	63.44±0.39	27.97±0.25	32.01±0.32	37.57±0.35
DKN	64.08±0.12	29.06±0.16	31.82±0.11	38.52±0.14	62.91±0.26	28.08±0.20	32.20±0.24	37.75±0.22
DAN	65.14±0.16	30.04±0.20	32.98±0.22	39.52±0.19	62.65±0.49	27.79±0.32	31.79±0.40	37.37±0.39
NAML	64.21±0.20	29.71±0.13	32.51±0.20	39.00±0.12	64.24±0.38	28.81±0.21	33.06±0.28	38.52±0.29
NPA	63.71±0.27	29.84±0.12	32.40±0.19	39.02±0.20	63.69±0.75	28.51±0.47	32.74±0.64	38.27±0.62
LSTUR	65.51±0.29	30.22±0.31	33.26±0.38	39.76±0.34	64.66±0.33	29.04±0.26	33.44±0.32	38.82±0.30
NRMS	65.36±0.21	30.02±0.11	33.11±0.15	39.61±0.14	65.15±0.13	29.29±0.12	33.78±0.13	39.24±0.13
KRED	65.61±0.35	30.63±0.27	33.80±0.24	40.23±0.27	65.47±0.07	29.59±0.04	34.15±0.05	39.69±0.05
FIM	64.46±0.22	29.52±0.26	32.26±0.24	39.08±0.27	65.67±0.20	29.83±0.24	34.51±0.31	39.97±0.25
KIM	**67.02**±0.14	**31.30**±0.26	**34.57**±0.32	**41.09**±0.25	**66.45**±0.13	**30.27**±0.09	**35.04**±0.09	**40.43**±0.12

4.2 Performance Evaluation

We compare *KIM* with several state-of-the-art personalized news recommendation methods, which are listed as follow: (1) *EBNR* [22]: representing user interest from user's click history via a GRU network [4]. (2) *DKN* [33]: applying a multi-channel CNN network [16] to embeddings of aligned words and entities in news titles to learn news representations. (3) *DAN* [47]: learning news representations from words and entities of news titles via a CNN network, and learning user interest representations via an attentive LSTM network [9]. (4) *NAML* [34]: learning news representations from news titles, bodies, categories, and sub-categories via multiple attentive CNN networks. (5) *NPA* [35]: using attention networks with personalized attention queries to learn news and user representations. (6) *LSTUR* [1]: modeling shot-term user interests from user's recent clicked news via a GRU network and modeling long-term user interest via user ID embeddings. (7) *NRMS* [38]: modeling news content and user click behaviors via multi-head self-attention networks. (8) *KRED* [19]: learning representations for news from the entities in news and their neighbors in the knowledge graph via a graph attention network. (9) *FIM* [32]: matching user and news from texts of users' clicked news and candidate news via CNN networks.

We repeat different experiments five times and list the average performance of different methods and corresponding standard deviations in Table 2. First, we can find *KIM* significantly outperforms other baseline methods which independently model candidate news and user interest without consideration of their relatedness, *LSTUR*, *NRMS* and *KRED*. This is because a user may be interested in multiple areas, and a candidate news may also contain multiple aspects and entities. Thus, it is difficult for these methods to accurately match user interest and candidate news since they are independently modeled in these methods. Different from these methods, in our *KIM* method we propose a knowledge-aware interactive matching framework to interactively model user interest and candidate news. Our *KIM* can effectively incorporate relatedness between clicked news and candidate news at both text and entity levels for better interest matching. Second, *KIM* also outperforms baseline methods which model user interest with the consideration of candidate news, such as *DKN*, *DAN*. This is because candidate news may cover multiple aspects, and a user may only be interested in a part of them [34, 35]. However, these methods model candidate news without the consideration of the target user, which may be inferior for further matching candidate news with user interest. Different from these methods, our *KIM* can model candidate news with target user information. Besides, in these methods clicked news and candidate news are also independently modeled from their content without consideration of their relatedness, which may be suboptimal for further measuring the relevance with candidate news and user interest inferred from clicked news. Different from these methods, in our *KIM* we propose a *knowledge co-encoder* and a *text co-encoder* to interactively learn knowledge-aware representations of both clicked news and candidate news.

4.3 Ablation Study

In this section, we conduct two ablation studies to evaluate the effectiveness of *KIM*. We first evaluate the effectiveness of different information, i.e., texts and knowledge, for news content modeling. Due to space limitation, we only show the experimental results on the *MIND* dataset in the following sections. The experimental results are shown in Fig. 5, from which we have several observations. First, removing texts seriously hurts the performance of *KIM*. This is because texts usually contain rich information on news content and are vitally important for news content understanding [45]. Removing texts makes the news representations lose much important information and cannot model news content accurately. Second, removing knowledge (i.e., entities and their neighbors in the knowledge graph) in news content modeling also makes the performance of *KIM* decline significantly. This is because textual information is usually insufficient to understand news content [19, 33]. 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 understanding user interest in candidate news. Thus, incorporating entity information into personalized news recommendation has the potential to improve the accuracy of recommendation.

Next, we evaluate the effectiveness of several important co-attention networks in *KIM* by replacing them with attention networks individually. Fig. 6 shows the experimental results, from which we have several findings. First, after removing the news

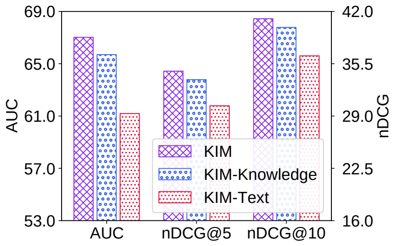

Figure 5: Performance of *KIM* with different information for news content modeling.

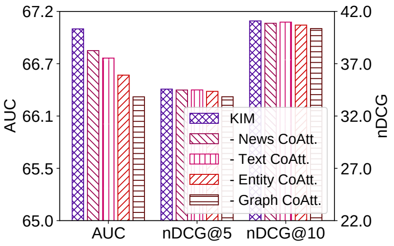

Figure 6: Ablation study on different components of *KIM*, where CoAtt. means the co-attention network.

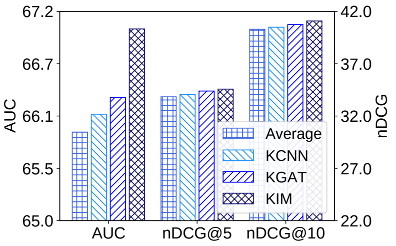

Figure 7: Performance of *KIM* and its variants with different knowledge modeling methods.

co-attention network in *user-candidate co-encoder*, the performance of *KIM* gets worse. This is because user interest may be diverse, and only a part of user's clicked news is informative for modeling the relevance between user interest and candidate news [33]. Besides, candidate news content may contain multiple aspects and a user may be interested in only a part of them. Thus, learning candidate news-aware user interest and user-aware 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 *KIM*. This is because semantic relatedness between clicked news and candidate news can help understand user interest in candidate news. Besides, a candidate news or a clicked news usually contains multiple aspects, and only a part of them is useful for the interest matching. Thus, it is difficult to effectively capture the relatedness of clicked news and candidate news at text level if their texts are independently modeled. Thus, interactively learning text-based representations of clicked news and candidate news via a text co-attention network can better capture relatedness between them for matching user interest with candidate news. Third, removing both the graph co-attention network and entity co-attention network makes the performance of *KIM* decline. This is because relatedness between clicked news and candidate news at entity level is also very informative for interest matching. Besides, it is also suboptimal for interest matching if the method represents clicked news and candidate news from their entities independently. In *KIM* method, both the graph co-attention network and entity co-attention network are used to capture relatedness between entities of clicked news and candidate news in an interactive way, which can incorporate rich information into *KIM* model for interest matching.

4.4 Effectiveness of Knowledge Modeling

We evaluate the effectiveness of the *knowledge co-encoder* in *KIM* by comparing *KIM* with its variations which independently model clicked and candidate news from their entities. The first one is *Average*, which averages embeddings of entities in news and their neighbors within K hops as the knowledge-based news representations. The second one is *KCNN*, which learns knowledge-based news representations from entities and their neighbors via the KCNN network proposed in *DKN* [33]. The third one is *KGAT*, which uses a knowledge graph attention network proposed in *KRED* [19] to learn knowledge-based news representations from entities in news and their neighbors on the knowledge graph. Besides, all of these variations have the same text modeling method with *KIM* for fair comparisons. Fig. 7 shows the experimental results.

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 conceptual 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, *KIM* significantly outperforms all of baseline methods, i.e., *Avg*, *KCNN*, *KGAT*. This is because relatedness between clicked news and candidate news at entity level can provide rich clues to infer user interests in candidate news. Besides, a clicked news or a candidate news may contain multiple entities and not all of them are useful for matching user interest with candidate news. However, these methods independently model entity information for clicked news and candidate news without consideration of their relatedness, which is suboptimal for further matching candidate news with user interest inferred from click history. Different from these methods, we propose a *knowledge co-encoder* to interactively learn knowledge-based representations for clicked news and candidate news from the relatedness between their entities for better interest matching.

4.5 Influence of Hyper-parameters

We evaluate the influence of an important hyper-parameter, i.e., the number of layers of the graph co-attention network, i.e., K, on the performance of *KIM*. Results are shown in Fig. 8, from which we have two observations. First, the performance of *KIM* first increases with the increase of K. This is because the relatedness between entities in clicked news and candidate news is informative for understanding user interest in candidate news. Besides, the GCAT network stacked for K layers can incorporate neighbors of entities in clicked news and candidate news within K hops for learning their representations. 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

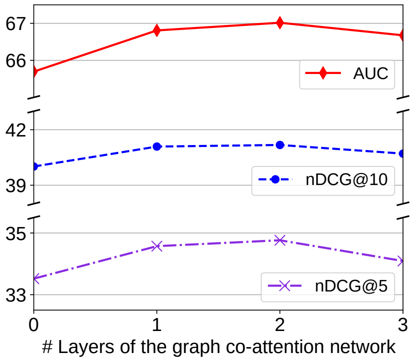

Figure 8: Model performance under different number of layers of the graph co-attention network, i.e., K.

accuracy. Second, when K is too large, the performance of *KIM* begins to decline. This is because when K becomes too large, too many multi-hop neighbors are considered when modeling the relatedness between user's clicked news and candidate news. This may bring much noise to the *KIM* model and hurt the recommendation accuracy. Thus, a moderate value of K, i.e., 2, is suitable for *KIM*.

4.6 Case Study

We conduct a case study to show the effectiveness of *KIM* by comparing it with *LSTUR* and *KRED*. We compare *LSTUR* since it achieves the best performance (Table 2) among baseline methods which model news content from pain news texts. Besides we compare *KRED* since it achieves the best performance (Table 2) among knowledge-aware baseline methods. 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 in Fig. 9, from which we have several observations. First, both *KRED* and *KIM* rank the candidate news clicked by the user higher than *LSTUR*. This is because it is difficult to understand the relevance of user interest and candidate news from the textual information of user's clicked news and candidate news. However, since Miley Cryus is a representative singer of country music, on the knowledge graph we can find that the entity "Country Music" in the first clicked news of the user has a link with the entity "Miley Cryus" in the candidate news clicked by the user. Thus, based on the information provided by the knowledge graph, *KRED* and *KIM* can better understand the relevance of user interest and candidate news. Second, *KIM* 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" also 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. However, it is difficult for *KRED* which independently model user's clicked news and candidate news to accurately capture the useful relatedness between entities of clicked news and candidate news for interest matching. Different from *KRED*, *KIM* uses a *knowledge co-encoder* to interactively represent clicked news and candidate news from their relatedness at entity level, which can better capture user interest in candidate news than *KRED*.

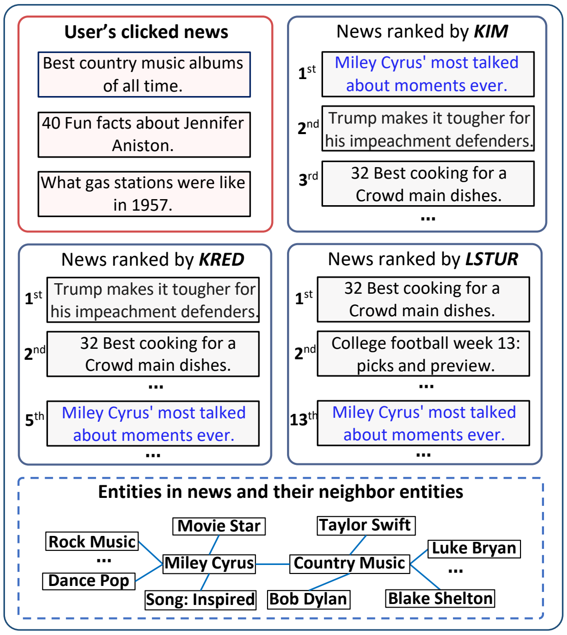
User's clicked news
Best country music albums of all time.
40 Fun facts about Jennifer Aniston.
What gas stations were like in 1957.
News ranked by KIM
1st
Miley Cyrus' most talked about moments ever.
2nd
Trump makes it tougher for his impeachment defenders.
3rd
32 Best cooking for a Crowd main dishes.
...
News ranked by KRED
1st
Trump makes it tougher for his impeachment defenders.
2nd
32 Best cooking for a Crowd main dishes.
...
5th
Miley Cyrus' most talked about moments ever.
...
News ranked by LSTUR
1st
32 Best cooking for a Crowd main dishes.
2nd
College football week 13: picks and preview.
...
13th
Miley Cyrus' most talked about moments ever.
...
Entities in news and their neighbor entities
Rock Music
...
Dance Pop
Movie Star
Miley Cyrus
Song: Inspired
Taylor Swift
Country Music
Bob Dylan
Luke Bryan
...
Blake Shelton

Figure 9: An illustrative case of news recommended by different methods. The news in blue is the news clicked by the user in this impression.

5 CONCLUSION

In this paper, we propose a knowledge-aware interactive matching framework for personalized news recommendation (named *KIM*). The framework aims to interactively model candidate news and user interests for more accurate interest matching. More specifically, we first propose a graph co-attention network to model entities based on the knowledge graph by selecting and aggregating the information of their neighbors which are informative for interest matching. We also propose to use an entity co-attention network to interactively model clicked news and candidate news from relatedness between their entities. Besides, we propose to use a text co-attention network to interactively model clicked news and candidate news from semantic relatedness between their texts. Moreover, we propose a *user-candidate co-encoder* to learn candidate news-aware user representation and user-aware 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 *KIM* method can significantly outperform other baseline methods.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant numbers U1936208, U1936216, 61862002, and U1705261.

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] 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*.

[5] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *WWW*. 271–280.

[6] Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph enhanced representation learning for news recommendation. In *WWW*. 2863–2869.

[7] 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.

[8] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *SIGIR*. 355–364.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* (1997), 1735–1780.

[10] 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.

[11] 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.

[12] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*. 1746–1751.

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

[14] Michal Kompan and Mária Bieliková. 2010. Content-based news recommendation. In *International conference on electronic commerce and web technologies*. 61–72.

[15] 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.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* (1998), 2278–2324.

[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] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

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

[25] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-Preserving News Recommendation Model Learning. In *EMNLP: Findings*. 1423–1432.

[26] 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.

[27] 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.

[28] 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.

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[30] Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge.. In *AAAI*. 855–860.

[31] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*. 448–456.

[32] Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *ACL*. 836–845.

[33] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *WWW*. 1835–1844.

[34] 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.

[35] 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.

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

[37] 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.

[38] 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.

[39] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2020. Neural news recommendation with negative feedback. *CCF TPCI* (2020), 178–188.

[40] Chuhan Wu, Fangzhao Wu, Junxin Liu, and Yongfeng Huang. 2019. Hierarchical user and item representation with three-tier attention for recommendation. In *NAACL*. 1818–1826.

[41] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Clickbait Detection with Style-Aware Title Modeling and Co-attention. In *CCL*. 430–443.

[42] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. SentiRec: Sentiment Diversity-aware Neural News Recommendation. In *AACL*. 44–53.

[43] 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.

[44] Chuhan Wu, Fangzhao Wu, Xiting Wang, Yongfeng Huang, and Xing Xie. 2021. FairRec:Fairness-aware News Recommendation with Decomposed Adversarial Learning. In *AAAI*.

[45] 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.

[46] 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.

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