Joint Knowledge Pruning and Recurrent Graph Convolution for News Recommendation

Yu Tian[1], Yuhao Yang[2], Xudong Ren[1], Pengfei Wang[3], Fangzhao Wu[4], Qian Wang[1], Chenliang Li[1†]
[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, 430072, China
s.braylon1002@gmail.com; {xdren69,qianwang,cllee}@whu.edu.cn
[2]School of Computer Science, Wuhan University, Wuhan, 430072, China
yhyeung@whu.edu.cn
[3]Beijing University of Posts and Telecommunications, China
wangpengfei@bupt.edu.cn
[4]Microsoft Research Asia, China
wufangzhao@gmail.com

ABSTRACT

Recently, exploiting a knowledge graph (KG) to enrich the semantic representation of a news article have been proven to be effective for news recommendation. These solutions focus on the representation learning for news articles with additional information in the knowledge graph, where the user representations are mainly derived based on these news representations later. However, different users would hold different interests on the same news article. In other words, directly identifying the entities relevant to the user's interest and deriving the resultant user representation could enable a better news recommendation and explanation.

To this end, in this paper, we propose a novel **kno**wledge **p**runing based **r**ecurrent gr**a**ph convolutional network (named Kopra) for news recommendation. Instead of extracting relevant entities for a news article from KG, Kopra is devised to identify the relevant entities from both a user's clicked history and a KG to derive the user representation. We firstly form an initial entity graph (namely interest graph) with *seed entities* extracted from news titles and abstracts. Then, a joint knowledge pruning and recurrent graph convolution (RGC) mechanism is introduced to augment each seed entity with relevant entities from KG in a recurrent manner. That is, the entities in the neighborhood of each seed entity inside KG but irrelevant to the user's interest are pruned from the augmentation. With this pruning and graph convolution process in a recurrent manner, we can derive the user's both long- and short-term representations based on her click history within a long and short time period respectively. At last, we introduce a max-pooling predictor over the long- and short-term user representations and the seed entities in the candidate news to calculate the ranking score for recommendation. The experimental results over two real-world datasets in two different languages suggest that the proposed Kopra obtains significantly better performance than a series of state-of-the-art technical alternatives. Moreover, the entity graph generated by Kopra can facilitate recommendation explanation much easier.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recurrent Graph Convolution, Knowledge Pruning, News Recommendation

ACM Reference Format:
Yu Tian, Yuhao Yang, Xudong Ren, Pengfei Wang, Fangzhao Wu, Qian Wang, Chenliang Li. 2021. Joint Knowledge Pruning and Recurrent Graph Convolution for News Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462912

1 INTRODUCTION

With the rapid development of the Internet, online news sites have become an integrated part of people's life. Given the tremendous amount of news generated every day, it becomes increasingly difficult for a person to find what she likes. Obviously, news recommendation, which aims to rank news articles in terms of the user's interest, plays more and more important role in enhancing user experience.

In the past few years, many news recommendation solutions have been proposed [11, 14, 16, 20, 29]. The earlier works mainly focus on extracting news representations by using the textual information like titles and news content [1, 28]. Compared with commodities [24] and movies [3], the semantic space of a news article is much wider since the latter could involve many entities of different types. However, due to the lack of background knowledge underlying these entities, these solutions cannot well represent news with sufficient semantics. Recently, several efforts resort to exploiting the background information in knowledge graphs (KGs) to enhance news representation learning [13, 20, 21, 23]. By considering the entities mentioned in the title and abstract as *seed entities*, we can obtain *contextual entities* in a KG by searching the neighborhood of former ones. Enriched by these contextual entities from a KG, these solutions achieves promising recommendation performance [6, 17, 22, 25, 31, 32].

[⋆]Yu Tian, Yuhao Yang and Xudong Ren contributed equally to this research.
[†]Chenliang Li is the corresponding author.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462912

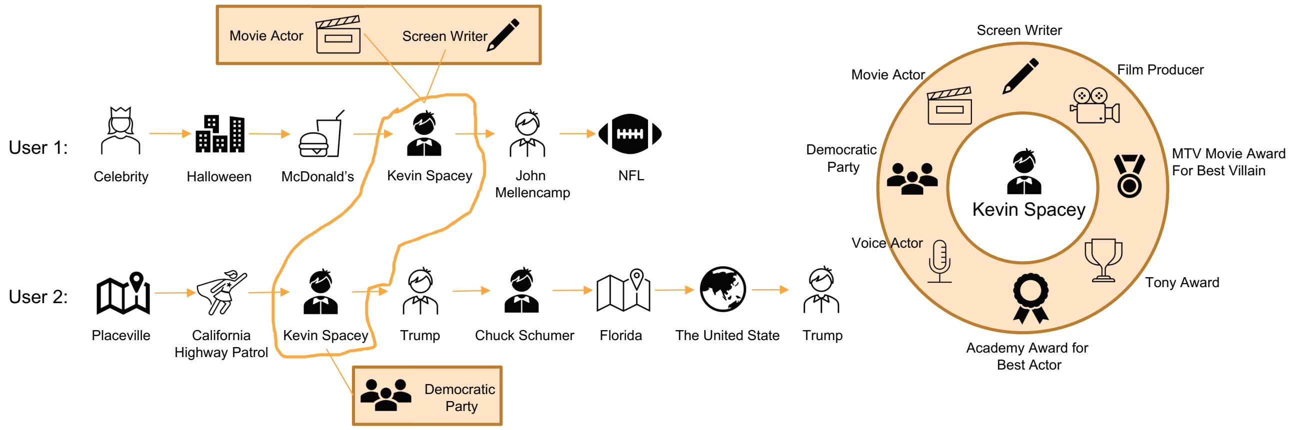

Figure 1: The sampled two users with their clicked news in terms of seed entities.

Despite the great success achieved by these KG-based solutions, they have two major limitations in common, which hinder the further performance improvement. *First*, all existing solutions overlook the direct user interest modeling via knowledge graph. They choose to perform representation learning for news articles by exploiting both the relevant information from the knowledge graph and news content in the first place. Then, a user encoder is utilized by compositing the representations of her clicked news, *e.g.,* attention mechanism [26, 27] or RNN [1, 15]. However, this post-processing treatment is inferior for user representation learning since the news representation is static and not tailored for each user. It is more effective to directly derive a user's representation from scratch.

Second, not all background knowledge provided by a KG is relevant to a user's interest. It is true that the contextual entities extracted from a KG could provide auxiliary information w.r.t. the entity mentioned in the news. But different users could hold different interests on the same news article. Figure 1 demonstrates two real users[1] and their click history in terms of seed entities from MIND dataset [30]. As shown in the right part of Figure 1, seed entity "Kevin Spacey"[2] [3] is associated with many other entities in Wikidata[4], covering his career as an actor and also his political views as a Democrat. Also, as shown in the left part of Figure 1, it is obvious that these two users hold totally different interests based on their clicked news. The first user is strongly interested in news related to entertainment, while the second user focuses more on the political events. That is, when learning the user's representation, we need to prune the knowledge provided by a KG such that the user's interest is well-matched. For example, entities "movie actor" and "screenwriter"for the first user, and "Democratic Party" for the second user.

To this end, in this paper, we propose a novel **kno**wledge **p**runing based **r**ecurrent graph convolutional network (named Kopra) for news recommendation. Kopra is devised to build an interest graph iteratively by including the relevant contextual entities along the sequence of the news clicked by the user. Firstly, we extract the entities from the news titles and abstract as *seed entities*. An initial graph is then formed over seed entities in terms of click order of corresponding news articles. Here, we design a recurrent graph convolution (RGC) to augment each seed entity recurrently with relevant contextual entities extracted from a KG. Specifically, for the current seed entity, we perform a RGC operation based on the graph updated until now to derive the representation of the current seed entity. Here, this representation is interest-aware since the current graph encodes the user interest to some extent. The contextual entities extracted from KG w.r.t. the current seed entity are then pruned based on the relevance between each contextual entity and the current seed entity. By including these relevant contextual entities in the current graph, we continue performing RGC for the next seed entity recurrently. After applying RGC over the last seed entity, we can produce an entity graph which reflects the user's interest precisely (namely interest graph), which would support for recommendation explanation. Also, the resultant user representation can precisely encode her interest for better news recommendation.

Note that the graph update with the relevant contextual entities is in discrete nature. Therefore, we perform the knowledge pruning and graph update through a reinforcement learning procedure. Given a user, we can utilize the above knowledge pruning and RGC to derive user's long-term and short-term representations over a long and a short history period respectively. At last, we develop a max-pooling predictor in Kopra by utilizing the long-term and short-term user representations and the seed entities in the target news to calculate the ranking score.

Empirically, we conduct extensive experiments on two real-world datasets (*i.e.,* MIND and Adressa) in two different languages. The results show that Kopra achieves substantial performance gains over state-of-the-art deep models for news recommendation. To summarize, we make the following contributions:

- We propose a novel knowledge-enhanced graph convolution model by pruning the irrelevant information of the knowledge graph for news recommendation. Also, instead of modeling news articles, we choose to utilize the entities

[1] The user ID is $U339891$ and $U497494$ respectively for the two users in the dataset.
[2] https://en.wikipedia.org/wiki/Kevin_Spacey
[3] Disclaimer: Kevin Spacey is mentioned for information purposes only, and his personal actions and opinions represent him only and have nothing to do with this article.
[4] https://www.wikidata.org/

mentioned in news to directly model user interest. To the best of our knowledge, this is the first attempt to explicitly and jointly perform knowledge pruning and user interest modeling for news recommendation.
- We introduce a novel recurrent graph convolution mechanism to help augment the seed entity sequence into an entity graph for each user. This graph is iteratively built with knowledge pruning and can encode the user's interest more comprehensively. Moreover, the graph works as a delegate to offer recommendation explanation.
- We perform extensive experiments on two real-world news datasets in two different languages. The results demonstrate the significant superiority of Kopra over the up-to-date yet compelling state-of-the-art technical alternatives. The knowledge graphs constructed for the two datasets will be released publicly for future research.

2 RELATED WORK

Personalized news recommendation is an important task in information retrieval field and has wide applications online. [7, 10] It is critical for news recommendation methods to learn accurate news and user representations.

In recent years, several deep learning methods have been proposed for personalized news recommendation, mainly based on attention mechanism to derive the user representation over the representations of their clicked news [14, 27, 28]. For example, Okura *et al.* propose to learn representations of news using denoising autoencoder and learn representations of users via a GRU network over their clicked news [14]. Wu *et al.* propose to learn news and user representations with personalized word- and news-level attention networks, which exploits the embedding of user ID to generate the query vector in the attention mechanism [27]. They also try to learn news representations from news titles by using multi-head self-attention [28]. The corresponding user representations are then learned with their click history via a multi-head self-attention. Their results suggest that the multi-head self attention mechanism can take any pairs of words or news into consideration, leading to better representation learning than CNN networks which cannot capture a broader range of contextual semantics.

However, these methods usually learn a single representation vector for each user, and cannot distinguish the long-term interests and short-term interests of users. Therefore, An *et al.* propose LSTUR to learn both long-term and short-term user representations with a GRU network. Their results suggest that both long-term and short-term user interests are effective to capture the diverse interests of users for better news recommendation [1].

Moreover, the semantic space for news articles is often much wider [16, 26]. Following this idea, some researchers have tried to use more information in a piece of news with a multi-view perspective. For example, Wu *et al.* propose NAML to learn representations of news and users by incorporating different kinds of news information such as title, body and topic category via an attentive multi-view learning framework [26]. However, these recent advances still follow the modeling methodology of deriving user interest over the news representations.

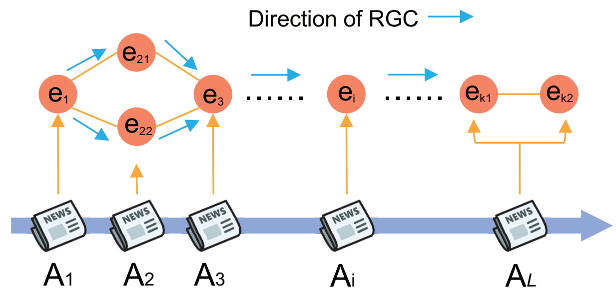

Figure 2: An interest graph initialized with seed entities. Arrowed lines indicate the convolution flow.

Another way to overcome the problem of rich semantic space is to exploit the external knowledge, usually with a KG [5]. For example, Wang *et al.* propose to learn representations of news from their titles via a knowledge-aware CNN network with the learned entity embeddings [21]. Similar to the above-mentioned methods, their user representations are also derived later based on their click history via an attention mechanism. Wang *et al.* propose a RippleNet model to combines embedding-based and path-based methods into a recommendation system based on the relations of a KG [20]. RippleNet chooses preference propagation in KG to continuously and automatically discover users' potential hierarchical interests. To enhance news representation learning, Lee *et al.* proposes TEKGR with a KG-level news encoder to construct a topic-enriched subgraph from the entities of news titles by adding their 2-hop neighbors with topical relations [8].

Different from these methods, our proposed Kopra can utilize a KG to model users' interest directly by pruning much irrelevant information of KG, leading to better user representation learning. A recurrent graph convolution is proposed to achieve knowledge pruning in an interest-aware manner. Extensive experiments on two real-world datasets validate the superiority of Kopra by achieving better performance on news recommendation.

3 METHOD

In this section, we firstly present a formal problem setting for our proposed Kopra. Then, we describe each component of Kopra in detail, followed by a reinforcement learning based optimization process.

3.1 Problem Formulation

Let $\mathcal{G} = \{(h, r, t)\}$ denote a knowledge graph, where knowledge triple (h, r, t) indicating there is a relation r held between head entity h and tail entity t. Here, we consider all entities that hold a relation with entity e as the contextual entities $C_e = \{x \mid \exists\, r : (e, r, x) \in \mathcal{G} \vee (x, r, e) \in \mathcal{G}\}$. For a given user u, we chronologically organize her clicked news articles as $\mathcal{H}_u = \{A_1, \ldots, A_L\}$, where L is the size of the history and A_L is the most recent news clicked by the user. Based on knowledge graph $\mathcal{G}$, we firstly extract entities $\mathcal{E}_{A_k}$ mentioned in the title and abstract of each news article A_k, which are considered as seed entities for the latter. Our task is to build an entity graph $\mathcal{G}_u$ to represent the user's interest effectively, namely interest graph, based on $\mathcal{G}, \mathcal{S}_u$. Then the news recommendation is to predict the likelihood that user u will read new v in terms of $\mathcal{G}_u$ and $\mathcal{E}_v$, where $\mathcal{S}_u = \{\mathcal{E}_{A_1}, \ldots, \mathcal{E}_{A_L}\}$.

3.2 Interest Graph Initialization

The temporal information has been proven to effective for many recommendation tasks. Hence, at the beginning, we choose to form an interest graph $\mathcal{G}_u$ from $\mathcal{S}_u$ by following the click order of these news articles. When only one seed entity is included in each $\mathcal{E}_{A_k}$, we can easily produce a sequence of seed entities as follows:

$$\mathcal{G}_u = \{\mathcal{E}_{A_1} \leftrightarrow \mathcal{E}_{A_2} \cdots \leftrightarrow \mathcal{E}_{A_L}\} \tag{1}$$

where symbol $\leftrightarrow$ refers to an edge connecting the two entities from the corresponding two sets. However, this strategy creates a problem that how to organize multiple entities mentioned in one article.

Actually, there are two scenarios when news has more than one seed entity: (1) these seed entities are near each other in knowledge graph $\mathcal{G}$. Here, we utilize the range of 2-hop neighborhood to check this proximity. In this scenario, it is reasonable to assume that the news talks about a specific topic. For example, a news containing two seed entities,"Houston Rockets"[5] and "Golden State Warriors"[6], suggests that this article is about NBA or basket crucial games. Therefore, we choose to connect these seed entities in terms of their appearance order in the title and abstract in a series way. Specifically, when $\mathcal{E}_k = \{e_{k1}, e_{k2}\}$ where e_{k1} appears before e_{k2}, the interest graph is updated as $\mathcal{G}_u = \{\mathcal{E}_{A_1} \cdots \leftrightarrow e_{k1} \leftrightarrow e_{k2} \leftrightarrow \ldots \mathcal{E}_{A_L}\}$. (2) When the seed entities have no obvious proximity to each other in the knowledge graph, *e.g.,* "National Weather Service" and "North Texas", each of them would be the reason why this user clicks this news. To enable Kopra to identify which entity refers to the user's interest, we organize them in a parallel mode. In this scenario, the interest graph is updated as $\mathcal{G}_u = \{\mathcal{E}_{A_1} \cdots \Leftrightarrow \{e_{k1}, e_{k2}\} \Leftrightarrow \ldots \mathcal{E}_{A_L}\}$, where symbol $\Leftrightarrow$ suggests that the set is fully connected. In Figure 2, we show an example of the interest graph initialized with these two scenarios.

3.3 Interest-aware Knowledge Pruning

After initializing the interest graph with $\mathcal{S}_u$, we can search relevant contextual entities for each seed entity in knowledge graph for information augmentation. This process is conducted following the sequential order of the user's clicked history in an interest-aware manner. That is, we iteratively add the relevant contextual entities of each seed entity as new nodes in $\mathcal{G}_u$.

Specifically, for seed entity e_k at the k-th step, we derive its representation in terms of $\mathcal{G}_u$ updated until then. Here, we utilize a recurrent graph convolution (to be detailed shortly) to derive the representation $\mathbf{h}_{e_k} = RGC(e_k, \mathcal{G}_u)$. Then, we search the knowledge triples that contain e_k as either a head entity or tail entity. In these triples, the other entities associated with e_k are taken as the contextual entities of e_k. For each contextual entity e, a perception layer $f_1(\cdot)$ is utilized to calculate the relevance between e and e_k as follows:

$$s(e_k, e) = f_1(\mathbf{h}_{e_k} \oplus \mathbf{x}_e) \tag{2}$$

[5] https://en.wikipedia.org/wiki/Houston_Rockets
[6] https://en.wikipedia.org/wiki/Golden_State_Warriors

where $\mathbf{x}_e \in R^{d_1}$ is the embedding vector for contextual entity e, $\oplus$ is vector concatenation operator, and d_1 is the dimension size. We retain the contextual entities that have the relevance larger than zero (*i.e.,* $s(e_k, e) > 0$) as the relevant contextual entities $\mathcal{R}_{uk}$ of e_k, and add them into $\mathcal{G}_u$ by connecting them to e_k. Following this procedure, we continue the knowledge pruning and augmentation for the following seed entities until the last one. The final graph can be used to derive the interest representation for user u. Note that we derive the interest-aware seed entity representation each time with new information updated in $\mathcal{G}_u$, which would lead to better knowledge pruning for the rest seed entities later. Recall that we can organize seed entities in a parallel mode (ref. Figure 2). For these parallel seed entities, we perform knowledge pruning and augmentation as well as RGC simultaneously. The whole pruning process is illustrated in the left part of Figure 3.

3.4 Recurrent Graph Convolution

RGC is devised to derive the interest-aware representation for each seed entity by taking the latter as central in $\mathcal{G}_u$. In detail, after knowledge pruning and graph augmentation for $(k-1)$-th seed entity e_{k-1}, we perform graph convolution to propagate the interest information to the current seed entity e_k in a bi-directional manner. In forward propagation, we perform graph convolution for each seed entity following the order of $e_1, \ldots, e_{k-1}$. Specifically, following the graph attention network (GAT) in [19], we perform recurrent graph convolution for current seed entity e_k as follows:

$$\mathbf{x}_{e_k} = \mathbf{x}_{e_k} + \sum_{e \in \mathcal{N}_{e_k}} \alpha(e, e_k)\mathbf{x}_e \tag{3}$$

$$\alpha(e, e_j) = \frac{\exp\left(LeakyReLU\left(\mathbf{w}_1^\top f_2(\mathbf{x}_e \oplus \mathbf{x}_{e_k})\right)\right)}{\sum_{e' \in \mathcal{N}_{e_k}} \exp\left(LeakyReLU\left(\mathbf{w}_1^\top f_2(\mathbf{x}_{e'} \oplus \mathbf{x}_{e_k})\right)\right)} \tag{4}$$

where $\mathcal{N}_{e_j}$ is entity neighbors of e_j in the current $\mathcal{G}_u$ excluding e_{j+1}, $\mathbf{x}_*$ is entity embedding, $\mathbf{w}_1 \in R^{d_2}$ is the attention weight vector, and $f_2(\cdot)$ is a full-connected layer. When a neighbor e is a contextual entity, $\mathbf{x}_e$ is set to be the entity embedding. In contrast, when the neighbor is a seed entity prior to e_j, the representation output previously by RGC is used instead: $\mathbf{x}_e = \mathbf{h}_e = RGC(e, \mathcal{G}_u)$.

Similarly, in the backward propagation, we perform graph convolution following the order of $e_n, \ldots, e_{k+1}$. This time, no relevant contextual entity is involved but just aggregates the more recent seed entities to e_k. Afterwards, the representation $\mathbf{h}_{e_k}$ of RGC is then derived by performing GAT over all neighbors of seed entity e_k in $\mathcal{G}_u$. The workflow of RGC and its network structure are illustrated in Figure 4.

3.5 Max-Pooling Predictor

After knowledge pruning and augmentation for all seed entities, the resultant interest graph can be used to derive the user representation. Since the interest information is aggregated along the chain of seed entities, we can simply take $\mathbf{h}_{e_M}$ as the user representation $\mathbf{h}_u$, where $M = |\mathcal{S}_u|$. However, recall that some seed entities could be organized parallelly. This could also happen for the most recent news A_L. Hence, we add a *pesudo* seed entity p in

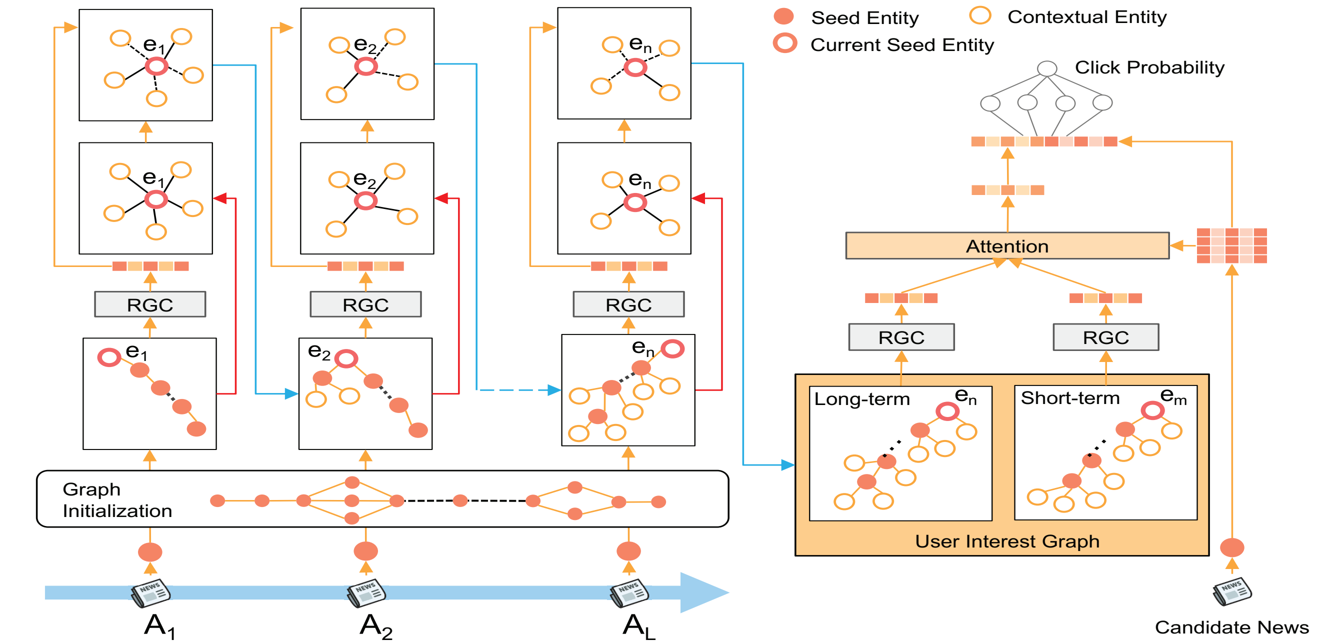

Figure 3: The overall framework of Kopra model (*Best viewed in color*).

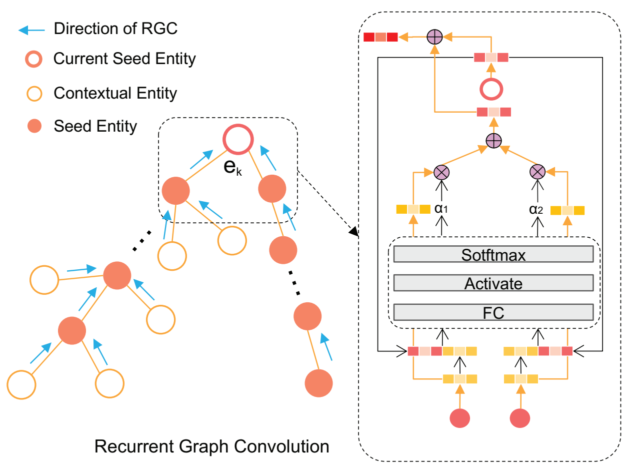

Figure 4: An illustration of recurrent graph convolution. Arrowed lines present flow of convolution.

the end to connect the last seed entity (set). The user representation is equivalent to the interest-aware representation generated by RGC for p: $\mathbf{h}_u = RGC(p, \mathcal{G}_u)$. Note that modeling both long- and short-term user interests is validated to be effective for sequential recommendation [1, 9]. Here, the user representation derived with L news articles above can be considered as long-term interest $\mathbf{h}_u^l$. We further obtain a short-term user representations $\mathbf{h}_u^s$ with recent S news only (*i.e.*, $S < L$).

Given a candidate news v with the corresponding seed entities $\mathcal{E}_v$, a user possibly clicks on this news because of one specific seed entity or because of semantics composited by these seed entities together. Accordingly, we form a matrix $\mathbf{H}_v$ to represent the multi-faceted information of the candidate news: $\mathbf{H}_v = [\mathbf{e}_{v_1}, \ldots, \mathbf{e}_{v_q}, \mathbf{h}_v]$, where $q = |\mathcal{E}_v|$, $\mathbf{e}_k$ is the embedding of k-th seed entity in $\mathcal{E}_v$ and $\mathbf{h}_v$ is the summation of these seed entity embeddings. Here, matrix $\mathcal{E}_v$ can be considered as multiple representations for the same news. For each news representation in $\mathbf{H}_v$, we utilize an attention mechanism to derive the final user representation as follows:

$$\mathbf{h}_u^k = \sum_{g \in \{s,l\}} \beta(v, g)\mathbf{h}_u^g \tag{5}$$

$$\beta(v, g) = \frac{\exp\left(\mathbf{w}_2^{\mathrm{T}} tanh(\mathbf{W}_1 \mathbf{h}_u^g + \mathbf{W}_2 \mathbf{H}_v[k])\right)}{\sum_{g' \in \{s,l\}} \exp\left(\mathbf{w}_2^{\mathrm{T}} tanh(\mathbf{W}_1 \mathbf{h}_u^{g'} + \mathbf{W}_2 \mathbf{H}_v[k])\right)} \tag{6}$$

where $\mathbf{H}_v[k]$ is k-th column of $\mathbf{H}_v$, $\mathbf{w}_2 \in R^{d_3}$ is the attention weight vector, $\mathbf{h}_u^g$ is long- or short-term user interest representation ,$\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices. At last, we calculate the likelihood $\hat{y}_{uv}$ of user u clicking this candidate news through a max-pooling as follows:

$$\hat{y}_{uv} = \sigma\Big(max\Big(\{f_3(\mathbf{H}_v[k] \oplus \mathbf{h}_u^k) \mid 1 \le k \le q+1\}\Big)\Big) \tag{7}$$

where $f_3(\cdot)$ is a perception layer, and σ is chosen to be sigmoid activation.

3.6 Model Optimization

Loss Function. Since news recommendation can be formulated as a click prediction task, we utilize the cross-entropy as the loss function:

$$\mathcal{L}_{uv} = -y_{uv} log(\hat{y}_{uv}) - (1 - y_{uv}) log(1 - \hat{y}_{uv}) \tag{8}$$

where y_{uv} is the ground truth whether the user clicked on this news or not.

Reinforcement Learning. The core merit of Kopra is to derive the user interests by performing knowledge pruning. However, this pruning process is conducted as a discrete selection (ref. Equation 2). To facilitate the model learning in an end-to-end fashion, we can model the optimization process as a reinforcement learning. By following the reinforcement learning terminology [18], we define an action as the knowledge pruning result $\mathcal{R}_{uk}$ of each seed entity. Before performing an action, the resultant interest-aware representation $\mathbf{h}_{e_k}$ of the seed entity produced by RGC is considered as the state. As to the reward, we utilize the loss function defined in Equation 8 as the delayed loss. Note that instead of choosing reward which the higher is better, we directly utilize the prediction loss for the model learning, which is equivalent in nature. We need to emphasize that no immediate reward can be derived for each action until we obtain the user's long- and short-term representations. Therefore, we choose to apply Monte Carlo search to obtain the delayed loss for each action. Specifically, after finishing knowledge pruning and augmentation on seed entity e_k, we update interest graph $\mathcal{G}_u$ with $\mathcal{R}_{uk}$. Then, we continue the whole process from here until the end for K times. The resultant loss values are $\mathcal{L}^1_{uvk}, \ldots, \mathcal{L}^K_{uck}$ respectively, and we perform an average over them as the expected loss $\mathcal{L}_{uc}(k)$ for action $\mathcal{R}_{uk}$:

$$\mathcal{L}_{uv}(k) = \frac{1}{K}\sum_{j}^{K}\mathcal{L}_{uvk}(j) \tag{9}$$

By aggregating the expected loss for all seed entities, we rewrite the loss function of Kopra as follows:

$$\mathcal{L}_{uv} = \frac{1}{M}\sum_{t}\mathcal{L}_{uv}(t) \tag{10}$$

Since we need to derive both long- and short-term user representations to calculate the loss, the above procedure applies for the corresponding long and short user click history respectively.

Knowledge Graph Correction. In case of novel entities that never appear in the training set, and to exploit the semantic relations provided by the knowledge graph further, we utilize TransE [2] to model knowledge graph. The detailed loss function is as follows

$$d = ||h + r - t|| \tag{11}$$

$$\mathcal{L}_{KG} = \sum_{(h,r,t)\in\mathcal{G}} max(0, d_{(h,r,t)} - d_{(h,r,t)^-} + \epsilon) \tag{12}$$

where $||\cdot||$ is the L_1 norm, $(h, r, t)^-$ is a corrupted triple w.r.t. (h, r, t) by replacing the head entity or the tail entity with another random one, and ϵ is the margin. We adopt *Adam* to perform parameter update by alternately learning Kopra and TransE. Our experimental results suggest that this knowledge graph learning could enhance the model generalization (ref. Section 4.5).

4 EXPERIMENTS

In this section, we evaluate our *Kopra* against the existing seven state-of-the-art solutions for news recommendation, including a series of ablation study, model analysis and case study.

Table 1: Statistics on the two datasets. # SE. per News: average number of seed entities per news.

Stats.	MIND	Adressa
# Users	1, 000, 000	3, 083, 438
# News	161, 013	48, 486
# Behaviors	24, 155, 470	27, 223, 576
# Entity per News	10.2	16.9
# SE. per News	2.5	7.8
# News per User	103.6	4.0
	Knowledge Graph	
# Relations	916	907
# Entities	43, 379	47, 991
# Triples	650, 346	855, 945

4.1 Datasets

Here, we utilize two real-world news recommendation datasets in different languages (*i.e.,* English and Norwegian) for evaluation.

MIND. The MIND dataset for news recommendation was collected from anonymized behavior logs of Microsoft News website [30]. In total, one million users who had at least 5 news clicks during six weeks (*i.e.,* October 12 to November 22, 2019) were randomly sampled. This is the largest English news recommendation dataset publicly available, which has already been used in [30].

Adressa. The Adressa Dataset is a Norwegian news dataset that includes more than 48K news articles and 3M anonymized users respectively [4]. The corresponding user behaviors are from Jan. 1 to Jan. 7, 2017. Here, we partition the dataset according to *activeTime* attribute, *i.e.,* the browsing time of the user on the news. In detail, when a browsing action takes a time less than 50-th percentile of all browsing times in the dataset, then it is labelled as negative.

Knowledge Graph Construction. We use knowledge triples from Wikidata as our KG. Note that both MIND and Adressa datasets provide with the entity information: from titles in Adressa and both titles and abstracts in MIND. We use *spacy-entity-linker*[7] to link these entities to Wikidata, and also extract and link entities from contents of the news articles.

As for Adressa, we observe that around 70% of provided entities cannot be matched back to Wikidata. Since *spacy-entity-linker* cannot perform Norwegian entity linking, we choose SpaCy to firstly perform entity recognition from titles, abstracts and contents as well. Then, we utilize the *Wikidata query*[8] to collect all the knowledge triples. Afterwards, we manually annotate the entities and link them back to KG for dozens of news articles where no entity is automatically extracted or linked successfully. It is worthwhile to note that the entities extracted from news contents are only utilized to clean KG and not used for news recommendation. Here, we adopt the work in [12] to perform the following processing: 1) adding a new group of entities for encoding topic context information; 2) adding relations between entities based on users' click behaviors and co-occurrence in news articles; and 3) we remove the triples with entities which do not appear in the news, including the news

[7] https://github.com/egerber/spaCy-entity-linker
[8] https://query.wikidata.org/

Table 2: Performance comparison of different models in MIND dataset. Symbol $*$ indicates that difference to the best is statistically significant at 0.05 level.

Model	Overall				Overlap Users				Unseen Users			
	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
DKN	64.02*	30.88*	33.64*	39.28*	64.92*	31.01*	33.83*	38.99*	63.89*	31.23*	33.57*	39.19*
NPA	66.37*	32.42*	34.70*	40.42*	67.05*	32.16*	35.07*	40.66*	66.01*	32.37*	34.73*	40.50*
NAML	66.83*	32.59*	35.20*	40.73*	67.16*	32.42*	35.33*	40.87*	66.87*	32.74*	35.36*	40.85*
LSTUR	67.58*	32.79*	35.40*	41.18*	67.99*	32.64*	35.63*	41.25*	67.39*	32.88*	35.58*	41.21*
NRMS	67.61*	33.02*	35.82*	41.49*	68.10*	33.14*	35.99*	41.67*	67.35*	33.09*	35.91*	41.47*
RippleNet	62.35*	29.96*	31.88*	37.65*	50.01*	25.12*	26.01*	33.17*	62.33*	29.93*	31.86*	37.63*
TEKGR	67.79*	33.31*	36.44*	41.98*	68.79*	33.30*	38.03*	42.91*	67.14*	32.79*	35.31*	41.18*
KOPRA	**68.80**	**34.64**	**41.59**	**44.89**	**69.57**	**35.12**	**41.63**	**45.46**	**68.21**	**33.71**	**40.90**	**44.68**

content. The resultant KG is then used for our experiments[9]. The detailed statistics of the two datasets are reported in Table 1.

4.2 Baselines

We compare our model with the following three KG-aware models and four KG-free alternatives.

KG-aware models. For models that exploit knowledge graphs to enrich semantic information, we choose DKN [21], RippleNet [20] and TEKGR [8].

- **DKN** utilizes convolutional neural networks to derive news representations over the textual words and entities in the news titles. The user representation is then derived with an attention over the click history.
- **RippleNet** simulates the propagation of water ripples by using the items that users are interested in as seeds. These seed items are then spread out to other items following the knowledge triples in KG. Here, we consider the seed entities as interested items for news recommendation.
- **TEKGR** utilizes three encoders on news title to extract representations in terms of textual words, topical information and contextual information from a KG respectively. These representations are then merged as the news representation. After that, an attention network is also utilized over the clicked news to derive user representation. TEKGR achieves promising performance by exploiting both topical information among entities and the contextual entities in KG.

KG-free models. Many KG-free models are also proposed for news recommendation. Here, we take NPA [27], NAML [26], LSTUR [1] and NRMS [28] as baselines, which mainly exploit auxiliary information in a multi-view perspective.

4.3 Experiment Setup

To ensure fair comparison, we use *Microsoft Recommenders* environment to evaluate all the models. As for MIND dataset, we closely follow the setting in [30]. That is, we randomly sampled half of the users for training, and the other half of the users are unseen users. There is also a validation set provided in the original dataset. In the testing phase, we separately use the training users (namely *Overlap Users*), the unseen users and all the users (namely *Overall*) as the test sets. This is to simulate the practical news recommendation scenario where unseen users are always emerging.

For experiments on Adressa dataset, we split the behaviors into sessions based on attributes *sessionStart* and *sessionEnd*. Then, these sessions are formatted the same as that of MIND dataset. Finally, we evaluate all models also on *Microsoft Recommenders* environment. Here, we use the first six days as the training set and the last day for testing. The validation set is taken as 20% of the training set.

Evaluation Metrics. The metrics used in our experiments are AUC, MRR, nDCG@5 and nDCG@10, which are standard metrics for recommendation evaluation. Each experiment is repeated ten times, and the results are averaged. For each model, we choose the optimal hyperparameter settings in terms of AUC on the validation set. The student $t-test$ is conducted for statistical significance test.

Hyper-Parameters. As to our Kopra, the following settings are used. The learning rate is 0.001, and margin ϵ is set to be 1. The dimension size for perception or full-connected layer is set to be 128 (*i.e.,* $d_3 = 128$). The embedding size d_1 is set to be 20 and 25 for MIND and Adressa respectively, and the corresponding $d_2 = 2 \cdot d_1$. Also, the long-term history size is 35 and 20 for the two datasets (*i.e.,* $L = 35/20$). We extract the short-term history as the most recent 25% and 20% of long-term history for MIND and Adressa respectively. Since a user would have a very short click history in total, in this scenario we pick the last news article as the short-term history. A pre-training is utilized for Kopra without knowledge pruning: three epochs are trained over the initialized interest graph with seed entities only. Moreover, we also use the early stop strategy.

4.4 Performance Evaluation

The performance comparison of different models in MIND dataset and Adressa dataset are reported separately in Table 2 and Table 3. Here, the following observations are made.

First of all, we observe that RippleNet consistently yields the worst performance across the two datasets. The possible reason is that there are many irrelevant contextual entities associated with the seed entity. By propagating through these noisy relations, the user interest could easily be corrupted. Similarly, DKN also performs significantly worse than the KG-free counterparts. This observation is consistent with what has been observed in earlier studies [1, 13, 30].

Second, across two datasets, the best KG-aware baseline TEKGR experiences significant performance decline on Adressa. For example, TEKGR outperforms all the KG-free models on MIND dataset

[9]The constructed knowledge graphs for MIND and Adressa along with our implementation of Kopra is available at https://github.com/WHUIR/KOPRA.

Table 3: Results on the test set of the Adressa dataset. Symbol $*$ indicates that difference to the best is statistically significant at 0.05 level.

Model	AUC	MRR	nDCG@5	nDCG@10
DKN	64.71*	60.29*	60.45*	67.43*
NPA	67.13*	62.42*	62.56*	68.31*
NAML	69.16*	62.92*	62.08*	68.89
LSTUR	65.87*	60.82*	60.64*	68.97*
NRMS	68.05*	62.31*	61.85*	**69.42**
RippleNet	49.77*	43.65*	54.56*	61.54*
TEKGR	67.54*	63.01*	59.77*	63.98*
KOPRA	**71.02**	**65.90**	**63.18**	69.13

Table 4: Performance comparison for Kopra and its three variants. KGC: Knowledge Graph Correction. MNR: Multiple News Representations; KP: Knowledge Pruning.

Model	MIND (Overall)			
	AUC	MRR	nDCG@5	nDCG@10
Kopra	68.80	34.64	41.59	44.89
Kopra w/o KGC	67.70	32.76	35.29	40.99
Kopra w/o MNR	68.29	34.64	39.98	41.24
Kopra w/o KP	67.35	32.43	35.27	40.45

in most of the metrics, but on Adressa it is surpassed by NRMS and NAML instead. This would be reasonable if we consider the difference of entity quality for the two datasets. That is, most entities in MIND are linked to Wikidata, which can reflect the news content precisely. However, since we perform NER using spaCy from scratch and filter out entities excluded in KG. Hence, the retained entities can be as short and common as words like "bilde" which just means "photo" in Norwegian, which incurs lots of information loss. This observation is also consistent with prior work [8].

Finally, on MIND dataset, Kopra consistently outperforms these baselines on all metrics, and across three settings of testing users. The results on Adressa (ref. Table3), are also relatively consistent. Kopra performs significantly better than most baselines on all metrics, but only marginally outperformed by NRMS on nDCG@10. Note that as we discussed above, the entity extracted from Adressa is far beyond the perfect. But Kopra still obtains much better performance than KG-free models in terms of AUC and MRR.

4.5 Analysis of Kopra

Ablation Study. Reflecting the intuitions of Kopra, we use entities to represent a user based on her click history, and introduce knowledge graph to enrich this form of user modeling. The key components include the knowledge pruning, KG correction, and multiple news representations (ref. Section 3.5). Accordingly, we come up with three variants of Kopra, with knowledge pruning, KG correction, multiple news representations removed separately. By removing the knowledge pruning, we just utilize all contextual entities to update the interest graph for each seed entity. As to multiple news representations, we simply take the summation of seed entity embeddings $\mathbf{h}_v$ as the news representation. That is, only

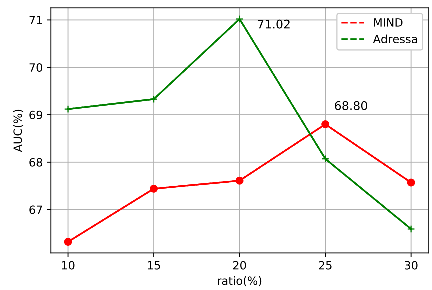

Figure 5: The AUC scores on MIND and Adressa with different ratios of long-term and short-term behaviors.

$k = q + 1$ is used in Equation 7 for prediction. The experimental results of the ablation studies in MIND for all users are reported in Table 4[10]. Here, we can make several observations:

1) The KG correction aims to propagate the representation change of the entities updated in each iteration to other related entities in KG. By this representation alignment with the background information provided in KG, Kopra can well avoid the model overfitting problem. For example, a news title could contain "Donald Trump" instead of "American President". But there is a very strong correlation for these two entities in KG (*i.e.,* ("Donald Trump", *position held*, "American President"). Without KG correction, the representation of entity "American President" may be unlikely to be updated, which may has a negative influence in the process of pruning and testing. Without KG correction, the performance reduction is up to 15.1% and 8.7% in terms of nDCG@5 and nDCG@10 respectively.

2) As for the multiple news representations, this design choice can well model the discrete and diverse semantics of a news article. By applying the max-pooling, the point and union interests can be well learned by Kopra. Without the multiple news representations, Kopra experiences a performance reduction up to 3.9% and 8.1% in terms of nDCG@5 and nDCG@10 respectively. Lastly, 3) it is clear that knowledge pruning largely improves the recommendation performance. The performance reduction without it is up to 15.2% and 9.9% in terms of nDCG@5 and nDCG@10 respectively.

Impact of Short-Term User Interest. Figure 5 plots the performance patterns by varying the ratio of short-term history size against long-term history size among {10%, 15%, 20%, 25%, 30%} for Kopra. In general, there is no perfect ratio value for both datasets. The best value is determined by the characteristics of user behaviors in each dataset. In MIND, a value of 25% can bring the best performance, while on Adressa the value is 20%.

4.6 Explainability Analysis

We further check whether Kopra can successfully discover both long- and short-term user interests by knowledge pruning. Here, we randomly choose one user, her history behaviors and a candidate news from MIND dataset[11]. Table 5 presents the title and abstract of the candidate news. The seed entities in it are highlighted in bold. This news is labeled as positive, indicating that the

[10] Due to space limitation, we omit the similar performance patterns in Adressa.
[11] The user and item is $U10550$ and $N104990$ respectively.

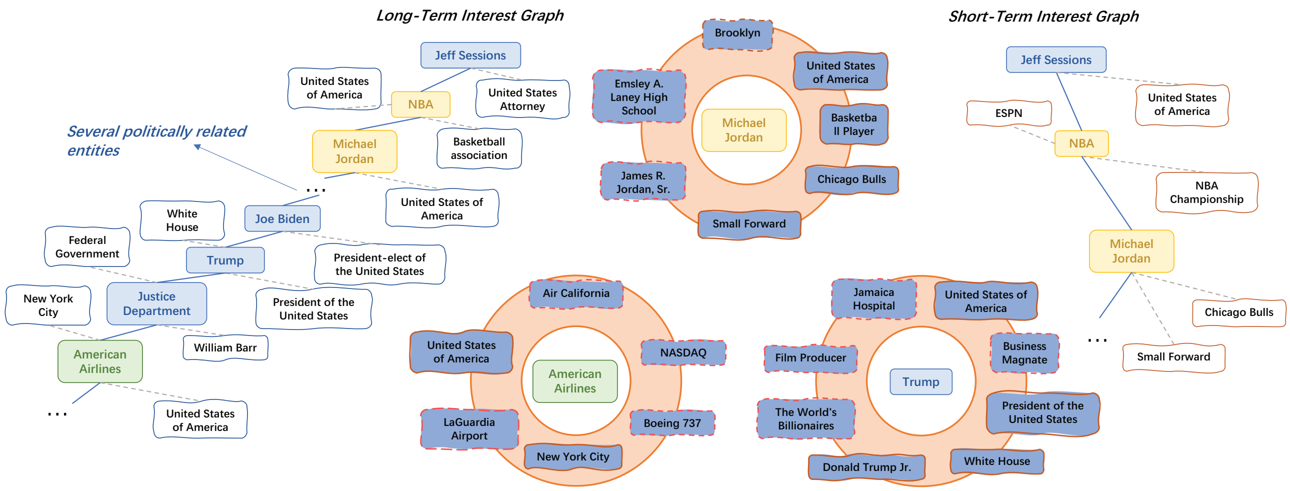

Figure 6: A case study with the user's long- and short-term interest graphs generated by KOPRA and some contextual entities for seed entity "Michael Jordan", "Trump" and "American Airlines" (*Best viewed in color*).

Table 5: Title and abstract of a candidate news in MIND.

***University of Florida** student president faces impeachment for **Trump Jr.**'s $50K campus talk.*
The University of Florida's student body president is facing calls for impeachment after paying Donald Trump Jr. $50, 000 in student tuition fees to speak on campus. Five student government senators allege president Michael Murphy abused his power when he brought Trump Jr. to campus for a speaking engagement last month. The impeachment resolution was first obtained by the Tampa Bay Times...

user will click it. For the sampled user, her partial long- and short-term interest graphs as well as three seed entities are illustrated in Figure 6. Note that the relevant contextual entities are connected to the corresponding seed entity in dashed line, for a more intuitive demonstration. Moreover, the background color in the rectangle of seed entities indicates their topic category. For example, the politics related entities have a light-blue color.

The long-term interest graph is shown in the left part. It is obvious that the main focus of this user is politics, while the short-term interest graph (in right part) shows her recent interest in sports. The middle three circles display some contextual entities of seed entity: "Michael Jordan", "Trump" and "American Airlines", where the contextual entities pruned by KOPRA are indicated with the dashed and sketched style. We can easily observe that the pruned contextual entities have nothing to do with the user long- or short-term interest. For example, the contextual entity "Film Producer" for "Trump" is neither related to politics nor to sports. In addition, our model treats pruning for short- and long-term interests differently. The pruning is performed in a user interest-aware fashion. Taking "Michael Jordan" as an example, in the long-term interest graph, the interest is biased towards national and political news, so the relevant contextual entity for "Michael Jordan" is only his country, *i.e.,* "United States of America". This indicates that the user focuses on "Michael Jordan" in the long-term because he is an American. But in the short-term interest graph, which implies the sports-related interest, the contextual entities retained by KOPRA are "Chicago Bulls" and "Small Forward". This observation suggests that our KOPRA is effective in capturing the user's interest in different granularities for better recommendation.

Given the candidate news is about politics and the long-term interest graph also reflects the same topic, the attention mechanism derives the correct importance weights, indicating the long-term user representation is more useful (0.5345 vs. 0.4655). Also, because the user focuses many named entities related to Trump in the long-term interest graph, she is likely to click the candidate news, which is about his child, and politics too. KOPRA echoes this interest with a click probability of 0.6654, suggesting that users will click on this news. However, we can also see that the given probability is not that high, because her recent interest (whose entities are in short term graph) falls more on the sports. Overall, our results demonstrate that the interest graphs generated by KOPRA can facilitate explanation with causes and effects at a finer level of granularity.

5 CONCLUSION

In this paper, our key focus is on the user interest modeling with relevant information provided by KG for news recommendation. The major idea is that not all contextual information of KG could be useful to understand users. Hence, we propose an interest-aware knowledge pruning, by introducing a novel recurrent graph convolution to capture the user's interest. Our extensive experiments demonstrate that better recommendation performance and user understanding are obtained compared with the existing SOTA alternatives. In the future, we plan to exploit the entities mentioned in the news content to further enhance the performance.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (61872278; 61802029; U20B2049; 61822207).

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 (1)*. 336–345.

[2] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.

[3] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *KDD*. 193–202.

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

[5] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A Survey on Knowledge Graph-Based Recommender Systems. *CoRR* abs/2003.00911 (2020).

[6] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *SIGIR*. 505–514.

[7] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. 2018. News recommender systems - Survey and roads ahead. *Inf. Process. Manag.* 54, 6 (2018), 1203–1227.

[8] Dongho Lee, Byungkook Oh, Seungmin Seo, and Kyong-Ho Lee. 2020. News Recommendation with Topic-Enriched Knowledge Graphs. In *CIKM*. 695–704.

[9] Chenliang Li, Xichuan Niu, Xiangyang Luo, Zhenzhong Chen, and Cong Quan. 2019. A Review-Driven Neural Model for Sequential Recommendation. In *IJCAI*. 2866–2872.

[10] Miaomiao Li and Licheng Wang. 2019. A Survey on Personalized News Recommendation Technology. *IEEE Access* 7 (2019), 145861–145879.

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

[12] Danyang Liu, Ting Bai, Jianxun Lian, Xin Zhao, Guangzhong Sun, Ji-Rong Wen, and Xing Xie. 2019. News Graph: An Enhanced Knowledge Graph for News Recommendation. In *KaRS@CIKM (CEUR Workshop Proceedings, Vol. 2601)*. 1–7.

[13] 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.

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

[15] Keunchan Park, Jisoo Lee, and Jaeho Choi. 2017. Deep neural networks for news recommendations. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2255–2258.

[16] T. Y. S. S. Santosh, Avirup Saha, and Niloy Ganguly. 2020. MVL: Multi-View Learning for News Recommendation. In *SIGIR*. 1873–1876.

[17] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent knowledge graph embedding for effective recommendation. In *RecSys*. 297–305.

[18] Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. 1999. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *NIPS*. 1057–1063.

[19] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *CoRR* abs/1710.10903 (2017).

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

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

[22] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In *KDD*. 968–977.

[23] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In *WWW*. 2000–2010.

[24] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba. In *KDD*. 839–848.

[25] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable Reasoning over Knowledge Graphs for Recommendation. In *AAAI*. 5329–5336.

[26] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Attentive Multi-View Learning. In *IJCAI*. 3863–3869.

[27] 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.

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

[29] 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.

[30] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A Large-scale Dataset for News Recommendation. In *ACL*. 3597–3606.

[31] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *KDD*. 974–983.

[32] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-Graph Based Recommendation Fusion over Heterogeneous Information Networks. In *KDD*. 635–644.