ICIC Express Letters
Part B: Applications
Volume **15**, Number **2**, February **2024**

ICIC International ©2024 ISSN 2185-2766
pp. 117–128

A KNOWLEDGE-GUIDED REINFORCEMENT LEARNING MODEL FOR NEWS RECOMMENDATION

Mei Zhang[1], Liangcai Li[2], Jiakai Li[1,2], Jianyong Duan[1,2,*], Xiao Yang[1,2]
and Qingsong Yuan[1,2]

[1]School of Information Science and Technology
[2]CNONIX National Standard Application and Promotion Lab
North China University of Technology
No. 5, Jinyuanzhuang Road, Shijingshan District, Beijing 100144, P. R. China
zm@ncut.edu.cn; liangcai.jeally@gmail.com; { 976942402; 1337920816 }@qq.com; qsong_yuan@126.com
*Corresponding author: duanjy@ncut.edu.cn

Received April 2023; accepted July 2023

Abstract. *Personalized news recommendation is becoming increasingly important for online news platforms to help users alleviate information overload and improve news reading experience. However, user-news interaction data is likely to be sparse, complicated and time-varying; it is essential to capture and predict future or long-term user preference for generating accurate recommendation over time. In this paper, we proposed a Knowledge-Guided Reinforcement Learning model (KGRL for short) for fusing knowledge graph information into an RL framework for news recommendation. Specifically, we use a modified transformer with category embedding to build news representation. Then we formalize the news recommendation task as a Markov Decision Process (MDP), and make three major technical extensions in this framework, including state representation, reward function and learning algorithm. First, we propose to enhance the state representations with KG information considering both exploitation and exploration. Second, we carefully design a composite reward function that is able to compute both sequence- and knowledge-level rewards. Third, we propose a new algorithm for more effectively learning the proposed model. Extensive experiment results on both next news and next session recommendation tasks show that our model can significantly outperform the baselines on MIND dataset.*

Keywords: News recommendation, Modified transformer, Knowledge graph, Reinforcement learning

1. **Introduction.**

Online news services such as news of MSN and Google which aggregate news from various sources and distribute them to a large population of users [1, 2]. An overwhelming amount of newly-sprung news is generated every day, making it difficult for users to seek for their interested news [3]. Thus, personalized news recommendation is very important for online news platforms to help users find their interested contents [4, 5].

There are two major problems in news recommendation: how to represent news articles which have rich textual content, and how to capture the dynamic changes of user interests to model user more accurately [6]. Various methods have been proposed to address this task, such as classic matrix factorization techniques [7] and popular neural network approaches [8, 9, 10, 11]. For example, Wang et al. [1] proposed DKN, which incorporates information from knowledge graph for better news recommendation. Specifically, DKN formed news representations from their titles and entities via Convolutional Neural Network (CNN). Then they utilized an attention network to select important clicked news for user representations. Typically, these deep learning methods show strong advantages

DOI: 10.24507/icicelb.15.02.117

in solving complex tasks and dealing with complex data, due to its capability to capture non-linear user-news relationships and deal with various types of data sources. It has thus been increasingly used in recommender systems. However, deep learning-based recommender systems have limitations in capturing interest dynamics [12] due to distribution shift, i.e., the training phase is based on an existing dataset which may not reflect real user preferences that undergo rapid change.

In contrast, deep reinforcement learning methods aim to train a model that can learn from interaction trajectories provided by the environment by combining the advantages of deep learning and reinforcement learning. Since an agent in RL can actively learn from users' real-time feedback to infer dynamic user preferences, RL is especially suitable for learning from interactions. However, a core concept or mechanism for RL models is the exploration process. It may not be reliable to adopt a blind or random exploration strategy for capturing the evolvement of user interests. Inspired by the availability of Knowledge Graph (KG) and its applicability in various fields [13], we would like to utilize the informative KG data to guide the RL-based learning method for news recommendation.

From all above, in this paper, we propose a novel **K**nowledge-**G**uided **R**einforcement **L**earning model (**KGRL** for short) for fusing KG information into an RL framework for news recommendation. Specifically, we use a modified transformer with category embedding to build news representation, and formalize the news recommendation task as a Markov Decision Process (MDP), and then make three major technical extensions in this framework. First, we propose to enhance the state representations with KG information. By learning both sequence-level and knowledge-level state representations, our model is able to capture user preference more accurately. Especially, we argue that it is important to utilize KG information in the exploration process. To achieve this, we construct an induction network that aims to predict future knowledge characteristics of user preference. In this way, we can learn knowledge-based user preference, considering both exploitation and exploration. Second, we carefully design a composite reward function that is able to compute both sequence-level and knowledge-level reward signals. For sequence-level reward, we borrow the BLEU metric from machine translation, and measure the overall quality of the recommendation sequence. For knowledge-level reward, we force the knowledge characteristics of the actual and the recommended sequences to be similar. Third, we propose a truncated policy gradient strategy to train our model. Concerning the sparsity and instability in training induction network, we further incorporate a pairwise learning mechanism with simulated subsequences to improve the learning of the induction network. To evaluate the proposed model, we construct extensive experiments on MIND dataset by comparing it with several competitive baselines. Experiment results on both next news and next session recommendation tasks show that our model can significantly outperform all the baselines in news recommendation tasks.

2. Related Work.

2.1. News recommendation. News recommendation has attracted increasing attention from both data mining and natural language processing fields. For example, Wu et al. [10] proposed NRMS, which learns news representation from news title by using multi-head self-attention to model the interactions between words and learn representations of users from their browsed history by using multi-head self-attention to capture their relatedness. Wu et al. [9] proposed an attentive multi-view learning framework to represent news articles from different news texts such as title, body and category. They used an attention model to infer the interest of users from their clicked news articles by selecting informative ones.

2.2. **Knowledge-based recommendation.** With the development of Knowledge Graph (KG) technology, researchers also try to combine KG to improve the performance of recommendation system. For example, Huang et al. [14] used memory networks to store and represent knowledge base information to improve the effectiveness and interpretability of sequential recommendation. Although it is effective to combine KG to improve performance, these works do not simulate the long-term interests of users, so performance may be limited.

3. **Our Approach.** In this section, we introduce the proposed Knowledge-Guided Reinforcement Learning model (KGRL) in detail, and the overall architecture of KGRL is presented in Figure 1. In what follows, we start with a Markov Decision Process (MDP) formulation for our task, introduce transformer based model for news representation and then present our extensions on state representation, reward function and learning algorithms.

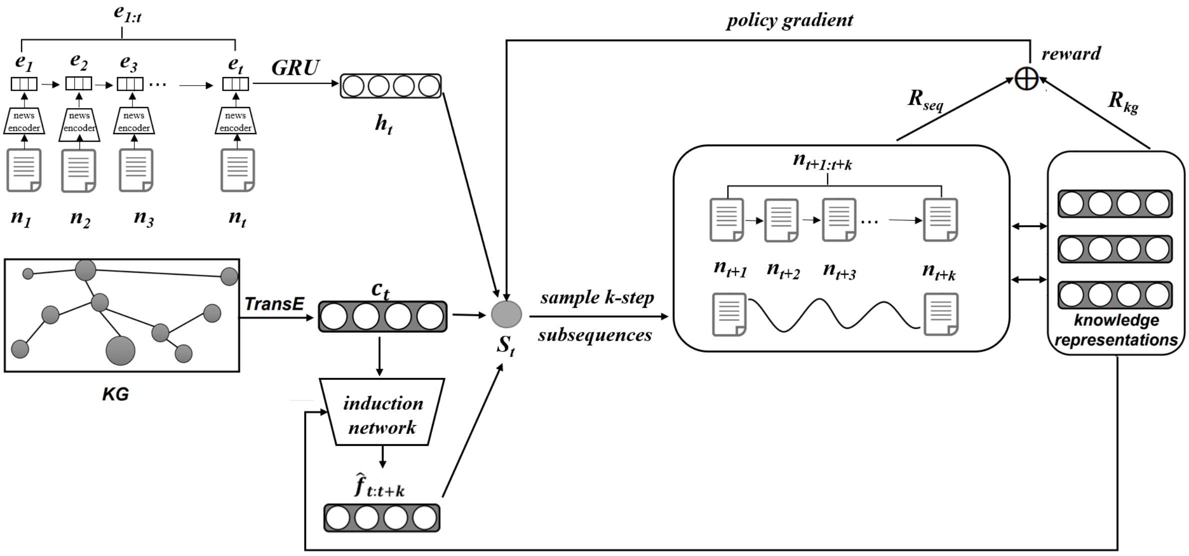

FIGURE 1. The overall architecture of Knowledge-Guided Reinforcement Learning model (KGRL). The news encoder is a modified transformer model, and three extensions are designed to fuse KG information to enhance the recommendation performance.

3.1. **An MDP formulation for news recommendation task.** We consider a Reinforcement Learning (RL) approach to news recommendation. We first briefly introduce MDP [15]. An MDP can be described by a quintuple $\langle \mathcal{S}, \mathcal{A}, T, R, \pi \rangle$: 1) $\mathcal{S}$ is a set of states, and each $s \in \mathcal{S}$ denotes the information state of the environment; 2) $\mathcal{A}$ is a set of actions, and each $a \in \mathcal{A}$ denotes an action that the agent that is able to perform; 3) T is the state transition function for updating the state according to the action and current state, i.e., $s_{t+1} = T(s_t, a_t)$; 4) R is the reward function and $r = R(s, a)$ giving the immediate reward of performing action a at state s; and 5) $\pi(a|s)$ describes the behavior of an agent, usually modeled by a probability distribution over the possible actions.

In order to frame the task, we use an MDP. In an MDP, there is an agent that interacts with the environment at discrete time steps. At each time step t, the process is in some state $s_t \in S$. In our task, the environment's state can be considered to include all useful information for sequential recommendation, including interaction history and KG. In our case, two major elements are considered:

$$s_t = [i_{1:t}, \mathcal{G}] \tag{1}$$

where $i_{1:t}$ represents the historical interaction information of the news that user u clicked before and $\mathcal{G}$ denotes the KG information. The initial state is set as $s_0 = [\emptyset, \mathcal{G}]$.

Following [16], we can use an embedding vector $v_{s_t} \in \mathbb{R}^{L_S}$ to encode the information of state s_t, and v_{s_t} is expected to encode useful information for representing state s_t.

In the current state s_t, the agent takes an action $a_t \in A$, which selects a news n_{t+1} from the candidate news set N for recommendation. Action behavior can be modeled by the policy $\pi(s_t)$, which defines a function that takes the state s_t as input and outputs a probability distribution over all possible clicked news. In this chapter, we use the softmax function to calculate the probability of likely clicking on the news:

$$\pi(a_t|s_t) = \frac{\exp\left\{e_{n_{j(a_t)}} W_1 v_{s_t}\right\}}{\sum_{n_j \in N} \exp\left\{e_{n_j} W_1 v_{s_t}\right\}} \tag{2}$$

where e_{n_j} represents the embedding vector of the jth news, W_1 is the parameter in the bilinear product, and v_{s_t} is the embedding vector of the state s_t.

After each action, the agent receives a numerical intermediate reward, i.e., $r_{t+1} = R(s_t, a_t)$. The reward function can be set to reflect the recommendation performance as needed in our task. Furthermore, it utilizes the transition function T $(T : \mathcal{S} \times T \to \mathcal{S})$ to update the state:

$$s_{t+1} = T(s_t, a_t) = T\left([u, e_{1:t}, G], e_{j(a_t)}\right) \tag{3}$$

The new state s_{t+1} can be written as $[e_{1:t+1}, G]$, which also associates an embedding vector $v_{s_{t+1}}$.

3.2. **Transformer for news representation.** The news encoder corresponds to the parallelogram box in Figure 1, and we apply the modified transformer for news representation which was motivated by Vaswani et al. [17]. The model is shown in Figure 2.

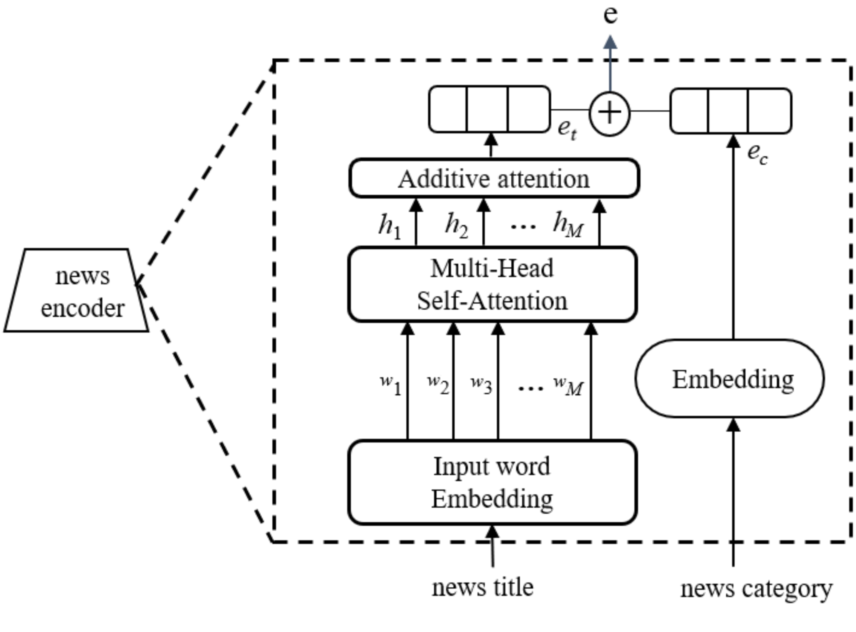

FIGURE 2. The modified transformer model for news representations

Although news headlines are typically concise and clear, we opt to simplify the transformer with a single layer of multi-head self-attention to avoid performance degradation caused by excessive parameters. This part consists of three layers. The first bottom layer is the word embedding layer, which is utilized to transform a news title from a sequence of words into a sequence of low-dimensional embedding vectors. Denoting a news title with M words, then through this layer it is converted into the embedded vector sequence

$[w_1, w_2, \ldots, w_M]$. The following layer is a word-level multi-head self-attention network. Interactions between words are important for learning news representations. Moreover, a word may relate to more than one word in the title. Thus, we employ the multi-head self-attention to form contextual word representations. The representation of the ith word learned by the kth attention head is computed as

$$\alpha_{i,j}^{k} = \frac{\exp\left(w_i^T W_s^k w_j\right)}{\sum_{m=1}^{M} \exp\left(w_i^T W_s^k w_m\right)} \tag{4}$$

$$h_i^k = W_v^k \left(\sum_{j=1}^{M} \alpha_{i,j}^{k} w_j \right) \tag{5}$$

where W_s^k and W_v^k are the projection parameters in the kth self-attention head, w is the word embedding vector and $\alpha_{i,j}^{k}$ indicates the relative importance of the interaction between the ith and jth words. The multi-head representation h_i^k of the ith word is the concatenation of the representations produced by h separate self-attention heads, i.e., $h_i = \left[h_i^1; h_i^2; \ldots; h_i^h\right]$. The total count of h_i is equal to the number of words M. To mitigate overfitting, we add dropout after self-attention.

The third layer is an additive word attention network to model relative importance of different words and aggregate them into title representations. Thus, we propose to use attention mechanism to select important words in news titles for learning more informative news representations. The attention weight β_i^w of the ith word is computed as

$$\beta_i^w = \frac{\exp\left(q_w^T \tanh\left(U_w \times h_i + u_w\right)\right)}{\sum_{j=1}^{M} \exp\left(q_w^T \tanh\left(U_w \times h_j + u_w\right)\right)} \tag{6}$$

where q_w, U_w and u_w are trainable parameters in the word attention network. The news title representation e_t is then calculated as

$$e_t = \sum_{i=1}^{M} \beta_i^w h_i \tag{7}$$

Since category information of user clicked news may also reveal their preferences, we model news categories via an embedding matrix, and denote the output of this embedding matrix as e_c. Then the final representation of the news is the concatenation of the title vector and the category vector, i.e., $e = [e_t; e_c]$.

3.3. Learning knowledge-enhanced state representation.

3.3.1. *Sequence-level status representation.* For the first state representation, a standard recurrent neural network is used to encode the historical interaction sequence, $n_1, n_2, \ldots, n_t$ represents the t news of the historical browsed record, and the vectors after the news encoder are $e_1, e_2, \ldots, e_t$, respectively. Then it is input to the GRU network as the input vector of each time step t, and the interest vector representation h_t of the user's final browsed history is obtained.

$$h_t = GRU\left(h_{t-1}, e_{n_t}; \Phi_{gru}\right) \tag{8}$$

where $GRU(\cdot)$ is the Gated Recurrent Unit, e_{n_t} is the embedding vector of news n_t and Φ_{gru} denotes all the related parameters of the GRU network. Such a representation mainly captures sequential characteristics of user preference, and it does not utilize knowledge information for deriving state representations.

3.3.2. *Knowledge-level state representation.* As shown in [14], KG data is useful to improve the performance of sequential recommendation algorithms. However, previous methods mainly consider enhancing news or user representations with KG data for fitting short-term behaviors with MLE. They seldom study how KG data can be utilized for exploration that optimizes long-term objective. To make a good trade-off between exploitation and exploration, we consider modeling two kinds of knowledge-based preference for a user, namely current knowledge-based preference (short as *current preference*) and future knowledge-based preference (short as *future preference*).

It is relatively easy to derive the current knowledge characteristics based on historical data. Recall that, each news article n_i is associated with some entities e_i in KG $\mathcal{G}$. The embedding vectors of these entities for news n_t trained by the TransE method are already given in the MIND dataset, denoted by $v_{e_{n_t}} \in \mathbb{R}^{L_E}$. Furthermore, we use a simple average pooling method to aggregate all the KG embeddings of the historical news that a user has interacted with:

$$c_t = \sum_{i=1}^{t} \text{Average}\left(v_{e_{n_t}}\right) \tag{9}$$

As the key point to achieve effective exploration, we incorporate *future preference* for capturing the possible interest evolving of a user at upcoming time steps. Intuitively, knowing future preference is useful for news sequential recommendation, especially in an RL setting. Based on current preference, our idea is to develop an induction network to directly predict the future preference. Specially, we construct a neural network using a multi-layer perception. At time step t, we predict a k-step future preference representation taking as input the current preference representation c_t in Equation (9):

$$f_{t:t+k} = MLP\left(c_t; \Phi_{mlp}\right) \tag{10}$$

where $f_{t:t+k}$ denotes the k-step future preference at time t, and we use Φ_{mlp} to represent parameters used in the induction network. Our assumption is that knowledge-based preferences tend to remain relatively stable over consecutive time steps. To predict future preferences based on existing information, our goal is to learn from KG data.

3.3.3. *Deriving the final state representation.* Based on the above discussions, we are ready to give the final state representation in our model. For a state s_t, its representation v_{s_t} is the combination of three representation vectors:

$$v_{s_t} = h_t \oplus c_t \oplus f_{t:t+k} \tag{11}$$

where "$\oplus$" is the vector concatenation operator, h_t is the sequence-level state vector in Equation (8), c_t is the current knowledge-based preference in Equation (9) and $f_{t:t+k}$ is the future knowledge-based preference in Equation (10). The first factor h_t mainly characterizes sequence-level information, the second factor c_t summarizes the existing knowledge characteristics for achieving knowledge-based exploitation, and the third factor $f_{t:t+k}$ predicts possible future preference for achieving knowledge-based exploration.

3.4. **Reward decomposition.** The interaction sequence is generated by the user's preference according to the news content of interest (which can be obtained from KG). Therefore, in addition to news article-level performance, it is also important to measure how good or bad the inferred knowledge-level preferences are. Based on the above description, at time step t, the k-step reward function is defined by integrating two different reward functions:

$$R\left(s_t, a_t\right) = R_{seq}\left(n_{t:t+k}, \hat{n}_{t:t+k}\right) + R_{kg}\left(n_{t:t+k}, \hat{n}_{t:t+k}\right) \tag{12}$$

Among them, $R_{seq}(\cdot, \cdot)$ and $R_{kg}(\cdot, \cdot)$ represent the sequence-level and knowledge-level rewards, respectively, and the subsequence $n_{t:t+k}$ and the recommended subsequence $\hat{n}_{t:t+k}$

are used as the information input. Note that only the input of k steps is considered here to approximate the overall performance. Next, we discuss how to calculate $R_{seq}(\cdot,\cdot)$ and $R_{kg}(\cdot,\cdot)$.

We borrow the metric of $BLEU$ for sequence recommendation. Formally, given actual interaction subsequence $n_{t:t+k}$ and the recommended subsequence $\hat{n}_{t:t+k}$, we define the reward function as

$$R_{seq}\left(n_{t:t+k}, \hat{n}_{t:t+k}\right) = \exp\left(\frac{1}{M}\sum_{m=1}^{M}\log prec_m\right) \tag{13}$$

where $prec_m$ is the modified precision and calculated as

$$prec_m = \frac{\sum_{p_m \in n_{t:t+k}} \min\left(\#\left(p_m, n_{t:t+k}\right), \#\left(p_m, \hat{n}_{t:t+k}\right)\right)}{\sum_{p_m \in n_{t:t+k}} \#\left(p_m, n_{t:t+k}\right)} \tag{14}$$

where p_m is the m-gram subsequence of the real click record $n_{t:t+k}$, and $\#\left(p_m, n_{t:t+k}\right)$ is the number of times that p_m appears in $n_{t:t+k}$. M determines how many M-gram precision values to use. It follows from the above formula that such a reward function supports the recommendation algorithm to generate more consistent m-grams from actual sequences. So it is natural to measure sequence-level performance in tasks.

In the second reward function, exact matches to news IDs are not concerned. Instead, we consider assessing the quality of knowledge-level features reflected in the sequence. Given an actual and predicted subsequence, i.e., $n_{t:t+k}$ and $\hat{n}_{t:t+k}$, still use the simple averaging method in Equation (9) to aggregate TransE, to derive subsequence-level knowledge representations, denoted by $c_{t:t+k}$ and $\hat{c}_{t:t+k}$, respectively. These two knowledge-level representations reflect users' preferences for the knowledge and logic behind the news. To measure the difference between two vectors, cosine similarity is used as the reward function:

$$R_{kg}\left(n_{t:t+k}, \hat{n}_{t:t+k}\right) = \frac{c_{t:t+k} \cdot \hat{c}_{t:t+k}^{T}}{\left|\left|c_{t,t+k}\right|\right| \cdot \left|\hat{c}_{t:t+k}\right|} \tag{15}$$

We can flexibly replace the cosine function with other forms of similarity measure. Then by plugging Equation (13) and Equation (15) into Equation (12), we can derive the final reward function. By providing these two reward signals, RL algorithms are able to produce better recommendation performance.

4. Experipences.

4.1. Experimental setup.

4.1.1. *Dataset.* In this paper, we conduct our experiments on the Microsoft News Dataset (MIND), which is a large-scale dataset for news recommendation research. It contains about 160k English news articles and more than 15 million impression logs generated by 1 million users. Every news article contains rich textual content including title, abstract, body, category and entities. Each impression log contains the click events, non-clicked events and historical news click behaviors of this user before this impression. The statistics of the MIND dataset are shown in Table 1.

Table 1. Statistics of the Microsoft news dataset

News	161,013	Users	1,000,000
News category	20	Impression	15,777,377
Click behavior	24,155,470	Avg. title length	11.52

4.1.2. *Baselines.* We adopt two types of baselines for comparison with our hybrid model, including sequential-based models, knowledge-based models. Here we do not compare to some deep learning models horizontally, such as DKN [1], LSTUR [8] and NRMS [10]. Because we are only exploring the effectiveness of integrating knowledge graphs under the setting of reinforcement learning.

- **GRU4Rec** is a session-based recommendation, which utilizes GRU to capture users' long-term sequential behaviors.
- **Ripple** is an embedding-based method that models users' potential interests along links in the knowledge graph for recommendation.
- **KGAT** explores the high-order connectivity with semantic relations in collaborative knowledge graph for knowledge-aware recommendation.

4.1.3. *Evaluation metrics.* The metrics used in our experiments are *AUC*, *MRR*, *nDCG*@10 and *hit-ratio*@10 (HR@10), which are standard metrics for recommendation result evaluation. Each experiment was repeated 10 times. The *AUC* indicator mainly measures the authenticity of the method, and its value range is $[0.5, 1]$. The larger the value, the higher the authenticity of the method. The *MRR* indicator mainly measures the ranking of the correct recommendation results in the recommendation list. The larger the value, the better the performance of the recommendation method. The *nDCG*@10 metric measures the relevance of the list of user recommendation results in the entire test set. The *hit-ratio*@10 metric measures the recall rate of correct news, the bigger the better.

4.1.4. *Parameter settings.* Set the batch size to 1024 and use the Adam optimizer to optimize all models. For the *BLEU* metric, consider up to 3-grams. For baseline models, each baseline is optimized against the validation set. For the KGRL model, the hidden layer size of GRU and the size of the vector representation of news through the news encoder are both set to 64, and the knowledge graph embedding vector size using TransE is 100 dimensions as given in the MIND dataset. For hyperparameters, the discount factor γ is set to 0.9 and the news segment length k is set to 3 and 5, respectively.

4.2. **Performance comparison.** In the next news recommendation and next session news recommendation tasks, this chapter compares the KGRL model with the competitive baseline methods mentioned above. The comparison results are given in Table 2.

Table 2. Performance comparison between the baseline model and KGRL

Task	Model	AUC	MRR	nDCG@10	HR@10
Next news recommendation	GRU4Rec	68.24	33.57	35.84	56.23
	Ripple	68.72	33.59	42.39	59.24
	KGAT	68.84	33.82	42.55	62.37
	KGRL	69.10	34.13	49.50	73.14
Next session recommendation	GRU4Rec	51.80	24.75	31.35	47.35
	KGRL	62.51	29.19	33.17	56.27

From Table 2, it can be observed that by integrating knowledge graph information into the recommender system, knowledge-based methods outperform order-based methods on all evaluation metrics. In particular, KGAT based on graph neural network can model higher-order connectives and achieve better performance than other knowledge-based methods (Ripple), and Ripple and KGAT are not provided in this chapter on session-based tasks results because they are not suitable for this task. For the hybrid model (i.e., knowledge+sequence), KGRL outperforms the other methods mentioned above, indicating that both sequence information and knowledge-level features of user clicks are useful for time-series recommendation.

4.3. **Ablation study.**

4.3.1. *Analysis on knowledge-enhanced state representation.* In this part we analyze the impact of fusing KG information into the state for news recommendation. Recall that we have three representations in Equation (11), namely h_t in Equation (8), c_t in Equation (9) and $f_{t:t+k}$ in Equation (10). Hence, we consider three variants for comparison by examining the effect of each part for sequential recommendation, including

- $KGRL_h$ using only the sequential representation h_t;
- $KGRL_{h+c}$ using the the sequential representation h_t and the current knowledge representation c_t;
- $KGRL_{h+f}$ using the sequential representation h_t and the future knowledge representation $f_{t:t+k}$.

The results of KGRL and its three variants on the MIND dataset are shown in Table 3. As can be seen from the table, $KGRL_h$ performs the worst on all evaluation metrics on the MIND dataset. This shows the necessity of fusing KG information into recommendation. Although adding future knowledge can improve recommendation performance (e.g., $KGRL_{h+f}$ outperforms $KGRL_h$), $KGRL_{h+c}$ still outperforms $KGRL_{h+f}$ on all evaluation metrics. Finally, by combining these three parts, the full model, i.e., KGRL, outperforms its three variants in all evaluation metrics.

Table 3. Performance comparison between $KGRL_h$, $KGRL_{h+c}$, $KGRL_{h+f}$ and KGRL

Task	Model	AUC	MRR	nDCG@10	HR@10
Next news recommendation	$KGRL_h$	67.76	33.05	41.63	53.61
	$KGRL_{h+c}$	67.99	33.36	42.13	68.75
	$KGRL_{h+f}$	67.83	32.88	41.52	57.96
	KGRL	69.10	34.13	49.50	73.14
Next session recommendation	$KGRL_h$	51.80	23.27	29.53	50.36
	$KGRL_{h+c}$	58.21	26.26	32.41	54.01
	$KGRL_{h+f}$	52.06	23.65	29.57	53.26
	KGRL	62.51	29.19	33.17	56.27

4.3.2. *Analysis on reward function.* In this section, the impact of each reward on the final performance will be analyzed. Specifically, according to Equation (12), two variants are introduced in Figure 3.

As can be seen, after convergence, $KGRL_{R_{seq}}$ outperforms $KGRL_{R_{kg}}$ on the nDCG@10 metric, while $KGRL_{R_{kg}}$ performs better on the Hit-Ratio@10 metric. This difference is likely caused by the nature of the two reward functions. The sequence-level reward optimizes the *BLEU* loss, tending to rank the correct news on higher positions, and the nDCG metric is the ranking of the item that measures the relevance in the recommendation list, so $KGRL_{R_{seq}}$ measured in nDCG@10 outperforms $KGRL_{R_{kg}}$. The knowledge-level reward optimizes the knowledge-based loss and tends to recall more correct news, so $KGRL_{R_{kg}}$ will outperform $KGRL_{R_{seq}}$ on the Hit-Ratio@10 metric. Combining their advantages, the overall model KGRL achieves better performance with these two reward functions.

4.3.3. *Analysis on the subsequence length.* To train the induction network, a k-step truncated learning strategy is employed to train sample sequences. Since KGRL looks k steps ahead to generate sample sequences, this paper explores the effect of exploring length k on the final performance through repeated experiments. Specifically, change the value of k in the set $k \in \{1, 3, 5, 7\}$. The test performance of KGRL on the MIND dataset with different lengths k is summarized in Table 4.

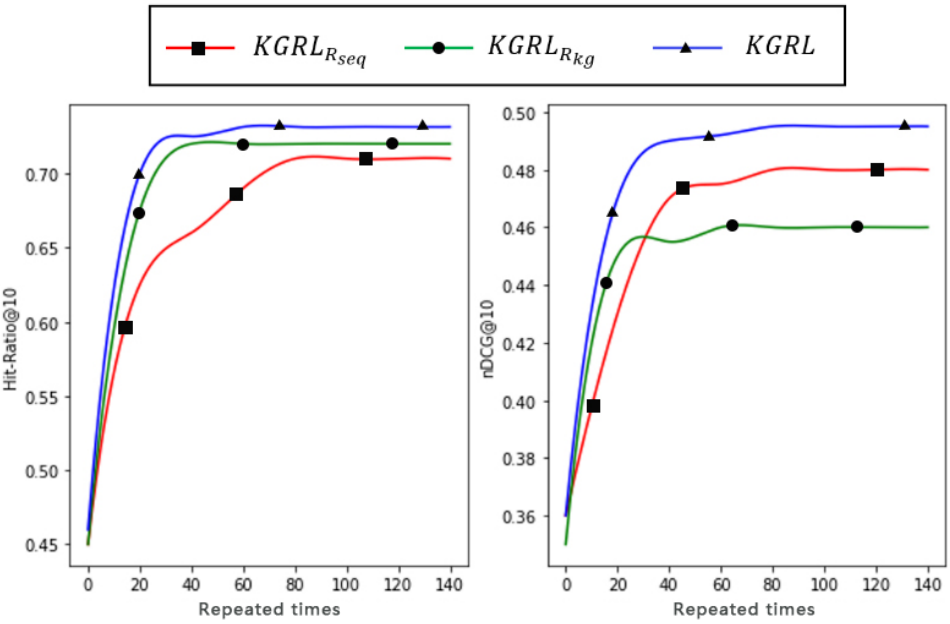

FIGURE 3. Performance curves of KGRL and its two variants with the varying iterations

TABLE 4. The performance impact of k-step truncated learning strategy

Task	k-step	AUC	MRR	nDCG@10	HR@10
Next news recommendation	$k = 1$	67.76	33.05	41.52	53.61
	$k = 3$	**68.16**	**34.13**	**49.50**	**73.14**
	$k = 5$	67.99	33.36	41.63	67.96
	$k = 7$	67.07	33.14	40.37	65.45
Next session recommendation	$k = 1$	51.84	22.71	28.53	50.36
	$k = 3$	52.80	23.27	29.46	51.53
	$k = 5$	**62.51**	**29.19**	**33.17**	**56.27**
	$k = 7$	58.76	27.56	30.43	54.10

The results in Table 4 show that KGRL achieves the best performance with $k = 3$ on the next news recommendation task and $k = 5$ on the next session recommendation task, because the latter task requires longer context to estimate future preferences. Finally, performance drops if longer exploration lengths are considered on both recommendation tasks. One possible reason is that longer exploration windows contain noisy information, which affects recommendation performance.

5. **Conclusions.** In this paper, we have presented a novel knowledge-guided reinforcement learning model, called KGRL, for fusing KG information into an RL framework for news recommendation. Specifically, we formalize the sequential recommendation task as a Markov Decision Process (MDP), and make three major technical extensions in this framework, including state representation, reward function and learning algorithm. In addition, we propose a new algorithm for more effectively learning the proposed model. The empirical results show that our model can significantly outperform the baselines on MIND datasets. For future work, we will consider how to adaptively learn better knowledge representation for news recommendation in the RL framework. Also, there may be some noise information in the interaction sequence or news titles. We will further study how to de-noise information in the news recommendation system.

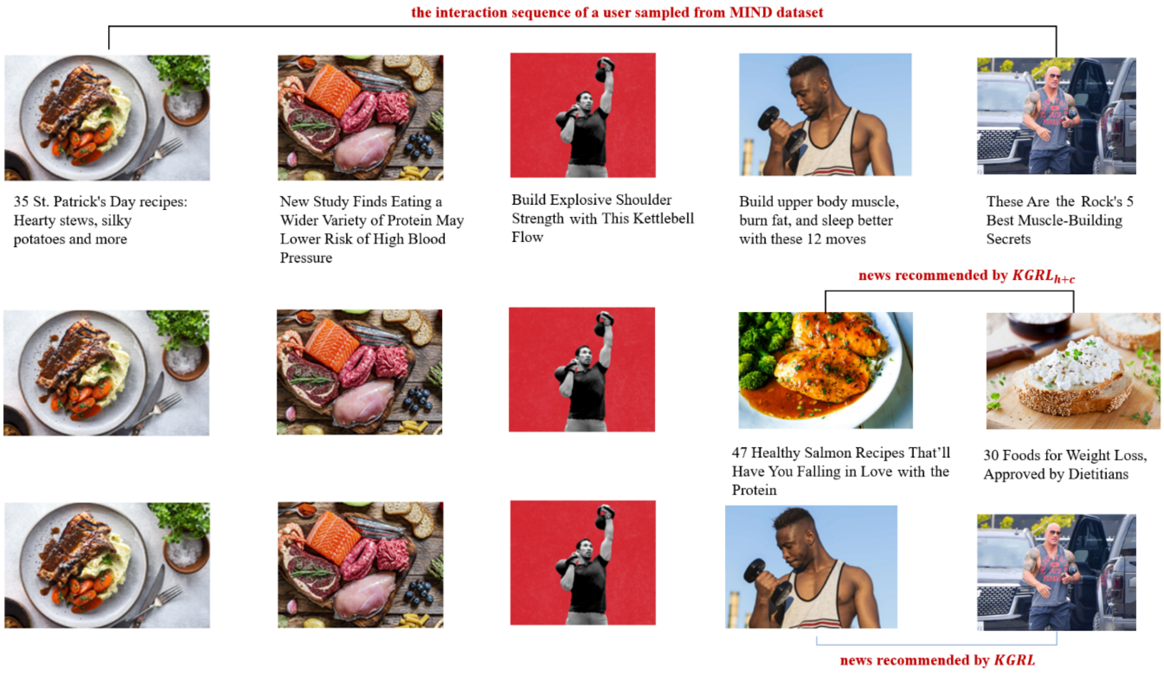
the interaction sequence of a user sampled from MIND dataset
35 St. Patrick's Day recipes: Hearty stews, silky potatoes and more
New Study Finds Eating a Wider Variety of Protein May Lower Risk of High Blood Pressure
Build Explosive Shoulder Strength with This Kettlebell Flow
Build upper body muscle, burn fat, and sleep better with these 12 moves
These Are the Rock's 5 Best Muscle-Building Secrets
news recommended by $KGRL_{h+c}$
47 Healthy Salmon Recipes That'll Have You Falling in Love with the Protein
30 Foods for Weight Loss, Approved by Dietitians
news recommended by $KGRL$

Figure 4. Case study on MIND dataset. The first line is the interaction sequence of the user, the last two news of other lines are the results of different recommendation methods.

Acknowledgment. This work is supported by the Humanities and Social Science Foundation of the Ministry of Education (Grant No. 21YJA740052).

REFERENCES

[1] H. Wang, F. Zhang, X. Xie and M. Guo, DKN: Deep knowledge-aware network for news recommendation, *arXiv Preprint*, arXiv: 1801.08284, 2018.

[2] H. Zhang, H. Sun, B. Qi and Z. Shen, News recommendation system based on topic embedding and knowledge embedding, *Wuhan University Journal of Natural Sciences*, vol.28, no.1, pp.29-34, 2023.

[3] S. Ge, C. Wu, F. Wu, T. Qi and Y. Huang, Graph enhanced representation learning for news recommendation, *arXiv Preprint*, arXiv: 2003.14292, 2020.

[4] W. IJntema, F. Goossen, F. Frasincar and F. Hogenboom, Ontology-based news recommendation, *Proc. of the 2010 EDBT/ICDT Workshops (EDBT'10)*, New York, NY, USA, 2010.

[5] Y. Lin, Y. Liu, F. Lin, L. Zou, P. Wu, W. Zeng, H. Chen and C. Miao, A survey on reinforcement learning for recommender systems, *IEEE Transactions on Neural Networks and Learning Systems*, DOI: 10.1109/TNNLS.2023.3280161, 2023.

[6] S. Okura, Y. Tagami, S. Ono and A. Tajima, Embedding-based news recommendation for millions of users, *The 23rd ACM SIGKDD International Conference*, 2017.

[7] Y. Koren, R. Bell and C. Volinsky, Matrix factorization techniques for recommender systems, *Computer*, vol.42, no.8, pp.30-37, 2009.

[8] M. An, F. Wu, C. Wu, K. Zhang and X. Xie, Neural news recommendation with long- and short-term user representations, *The 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[9] C. Wu, F. Wu, M. An, J. Huang, Y. Huang and X. Xie, Neural news recommendation with attentive multi-view learning, *The 28th International Joint Conference on Artificial Intelligence*, 2019.

[10] C. Wu, F. Wu, S. Ge, T. Qi and X. Xie, Neural news recommendation with multi-head self-attention, *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[11] Q. Zhu, X. Zhou, Z. Song, J. Tan and L. Guo, DAN: Deep attention neural network for news recommendation, *Proc. of the AAAI Conference on Artificial Intelligence*, vol.33, pp.5973-5980, 2019.

[12] X. Chen, C. Huang, L. Yao, X. Wang, W. Zhang et al., Knowledge-guided deep reinforcement learning for interactive recommendation, *2020 International Joint Conference on Neural Networks (IJCNN)*, pp.1-8, 2020.

[13] Q. Zhang, Z. Sun, W. Hu, M. Chen, L. Guo and Y. Qu, Multi-view knowledge graph embedding for entity alignment, *arXiv Preprint*, arXiv: 1906.02390, 2019.

[14] J. Huang, W. X. Zhao, H. Dou, J.-R. Wen and E. Y. Chang, Improving sequential recommendation with knowledge-enhanced memory networks, *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp.505-514, 2018.

[15] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, *IEEE Transactions on Neural Networks*, vol.9, no.5, p.1054, DOI: 10.1109/TNN.1998.712192, 1998.

[16] Y. Feng, J. Xu, Y. Lan, J. Guo and W. Zeng, From greedy selection to exploratory decision-making: Diverse ranking with policy-value networks, *The 41st International ACM SIGIR Conference*, 2018.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, Attention is all you need, *arXiv Preprint*, arXiv: 1706.03762, 2017.