

# A Knowledge-Aware Attentional Reasoning Network for Recommendation

Qiannan Zhu,<sup>1, 2</sup> Xiaofei Zhou\*,<sup>1, 2</sup> Jianlong Tan,<sup>1, 2</sup> Li Guo,<sup>1, 2</sup>

<sup>1</sup>Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

<sup>2</sup>School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China  
zhuqiannan@iie.ac.cn, zhouxiaofei@iie.ac.cn

## Abstract

Knowledge-graph-aware recommendation systems have increasingly attracted attention in both industry and academic recently. Many existing knowledge-aware recommendation methods have achieved better performance, which usually perform recommendation by reasoning on the paths between users and items in knowledge graphs. However, they ignore the users' personal clicked history sequences that can better reflect users' preferences within a period of time for recommendation. In this paper, we propose a knowledge-aware attentional reasoning network KARN that incorporates the users' clicked history sequences and path connectivity between users and items for recommendation. The proposed KARN not only develops an attention-based RNN to capture the user's history interests from the user's clicked history sequences, but also a hierarchical attentional neural network to reason on paths between users and items for inferring the potential user intents on items. Based on both user's history interest and potential intent, KARN can predict the clicking probability of the user with respect to a candidate item. We conduct experiment on Amazon review dataset, and the experimental results demonstrate the superiority and effectiveness of our proposed KARN model.

## Introduction

Recently knowledge graphs (KGs) have made significant contributions on many fields, such as recommendation system (Zhang et al. 2018), question answering system (He et al. 2014) and information extraction (Cao et al. 2018), etc. KGs are directed heterogeneous graphs that represent nodes as entities and edges as relations. Numerous structured data are stored in KGs, where  $(h, r, t)$  denotes the relation  $r$  links the head entity  $h$  and tail entity  $t$ . Focusing on recommendation field, the user-item knowledge graphs provide the rich knowledge of users, items, and the relationships among them. For instance, the facts of the user-item interaction are organized in the form of triplets like  $(James\ Cameron, IsDirectorOf, Aliens)$ , which represents *James Cameron* is the director of the movie *Aliens*.

\*Corresponding Author

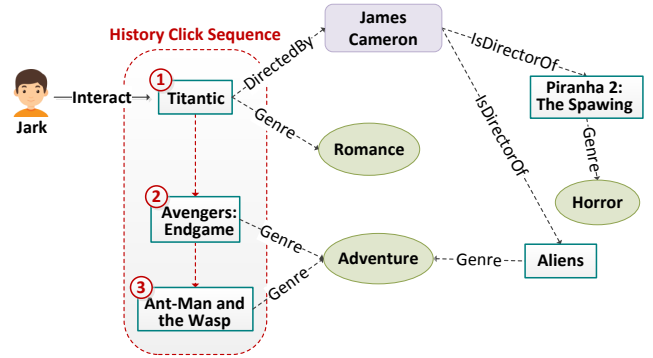


Figure 1: An example of knowledge-graph-aware recommendation in movie domain. The user's history click sequences are incorporated for more precise recommendation.

A large number of efforts have been done for making personalized recommendation, which roughly can be categorized collaborative filtering (CF) based methods (Das et al. 2007; Xue et al. 2017), content based methods (Ijntema et al. 2010; Zhang, Liu, and Gulla 2018) and hybrid methods (Li et al. 2011). These works analyze and process various content data such as users' profile, ratings, reviews, and items' attributes for personalized recommendation. Factually, numerous of structural information implied in these data can be used for modeling recommendation mechanism. In recent years, incorporating knowledge graphs into recommendation becomes very popular. The heterogeneous multi-type user behaviors and the item knowledge can be integrated into a unified graph structure, which has shown great potential in boosting the explainability of recommendation systems.

Among the knowledge-aware recommendation methods, the embedding-based (Wang et al. 2018b) and path-based (Wang et al. 2019) recommendation have successfully achieved promising recommendation performance. Importantly, the path fashion methods can better provide the ability of explainability for recommendation by searching the connectivity information of users and items in KGs. Giving an example of movie recommendation in Figure 1. From the path  $(Jark, Interact, Titanic) \Rightarrow$

(*Titantic*, *DirectedBy*, *James Cameron*)  $\Rightarrow$  (*James Cameron*, *IsDirectorOf*, *Piranha 2:The Spawning*), we can reason that the user *Jark* may interact the movie *Piranha 2:The Spawning* due to it has the same director *James Cameron* with *Titantic* he has watched before. The prior path-based methods (Zhao et al. 2017; Wang et al. 2019) only take advantages of path connectivity information between users and items, and ignore the importance of the user’s clicked history sequences on reflecting user’s preferences. As Figure 1 shown, without considering the users’ clicked sequences, the movie *Aliens* may be also recommended to user *Jark* along the reasoning path (*Jark*, *Interact*, *Titantic*)  $\Rightarrow$  (*Titantic*, *DirectedBy*, *James Cameron*)  $\Rightarrow$  (*James Cameron*, *IsDirectorOf*, *Aliens*). If we consider the user’s clicked history sequence *Titantic*  $\rightarrow$  *Avengers:Endgame*  $\rightarrow$  *Ant-Man and the Wasp* that reveals the user *Jark* prefers adventurous movies in recent times, the movie *Aliens* should be the more appropriate recommendation item than *Piranha 2: The Spawning* for user *Jark*.

In this paper, we propose a knowledge-aware attentional reasoning network denoted as KARN, which fuses the user’s clicked history sequence and path connectivity between users and items in KGs for recommendation. KARN is a deep learning model for click-through rate (CTR) prediction, which takes one user’s click history sequence and path connectivity between an user-item pair in KGs as input, and outputs the probability of the user clicking the candidate item. Specifically, for each item entity in user-item KGs, KARN first extracts item representation from the textual knowledge (i.e., its title) and contextual knowledge (i.e., its one-hop neighbors in KGs). Then KARN devises two key components with the goal of fully exploiting users’ click history sequence and path connectivity between users and items in KGs. The first component, namely SRA, that is the stack of the recurrent neural network and an attention network, aims to capture user’s history interest representation from the user’s clicked history sequence. The second component A-SRA first uses SRA component to obtain the single-path feature of each path between an user-item pair, and then employs the attention network to generate the potential user intent representation by aggregating multiple single-path features with different weights. Based on the representations of user’s history interest and potential intent, KARN calculates a probability to determine whether the user clicking the candidate item.

In summary, the contributions of this paper are as follows.

- We propose a knowledge-aware attentional reasoning network KARN that fuses the users’ clicked history sequence and path connectivity in KGs for recommendation.
- We extract the item representation by incorporating the textual knowledge and contextual knowledge of items based on the user-item knowledge graph.
- We present two components to respectively obtain the features of user’s history interest and potential intent from the user’s clicked sequence and path connectivity in KGs.
- We empirically perform extensive experiments on Amazon review data set. The experimental results demonstrate our model KARN significantly achieves state-of-the-art recommendation performance.

## Problem Formulation

This paper focuses on product recommendation, which usually provides two groups of correlation information for recommendation, i.e., the user-item interaction and item-item correlation. The user-item interaction represents the interaction behaviors between users and items, such as clicking, watching, purchasing, etc. The item-item correlation indicates the two items have the explicit relationships, such as bought together by the same user, belong to the same category, etc.

For the user-item knowledge graph construction, following (Catherine and Cohen 2016), we simply convert the user-item interaction and item-item correlation into following groups of triplet facts: (1) (*u*, *Purchase*, *v*) represents that the purchase interaction between the user *u* and the item *v*. (2) (*u*, *Review*, *v*) represents that the user *u* gives the feedback reviews on the item *v*. (3) (*v*, *BelongsTo*, *c*) represents that the item *v* belongs to the category *c*. (4) (*v*, *ProducedBy*, *b*) represents that the item *v* is produced by the brand *b*. (5) (*v*<sub>1</sub>, *AlsoBought*, *v*<sub>2</sub>) represents that items *v*<sub>1</sub> and *v*<sub>2</sub> have been purchased by the same user. (6) (*v*<sub>1</sub>, *AlsoViewed*, *v*<sub>2</sub>) represents that the second item *v*<sub>2</sub> was viewed before or after the purchase of the first item *v*<sub>1</sub>. (7) (*v*<sub>1</sub>, *BoughtTogether*, *v*<sub>2</sub>) represents that the items *v*<sub>1</sub> and *v*<sub>2</sub> have been purchased together in a single transaction. (8) (*v*<sub>1</sub>, *BuyAfterViewing*, *v*<sub>2</sub>) represents that the item *v*<sub>2</sub> has been purchased after viewing the reviews of item *v*<sub>1</sub>.

**Task Definition.** The goal of our work is to find the items that have the possible *Purchase* relationship with users. We can formulate the task as follows: given an user *u*, a candidate item *v*, the user’s history clicked item sequence  $S(u) = \{v_1, v_2, \dots, v_t\}$  and the set of paths  $P(u, v) = \{p_1, p_2, \dots, p_n\}$  between user *u* and item *v*, the holistic goal is to estimate the *Purchase* interaction probability between user *u* and item *v* by  $\tilde{y} = g(u, v | S(u), P(u, v))$ , where *g* is our recommendation system.  $\tilde{y}$  can also be explained that it aims to measure the plausibility of the triplet (*u*, *Purchase*, *v*) based on the user’s clicked sequence and the path connectivity.

## The Proposed Model

### Framework

We illustrate the framework of KARN in Figure 2. As shown in this figure, KARN takes the user’s clicked history sequences and path connectivity information between the user-item pairs as input, and outputs a probability of the user clicking the candidate item. It mainly consists of three steps: 1) Item representation extraction, 2) User’s history interest extraction and user’s potential intent extraction and 3) Recommended probability calculation. To be specific, in the first step, KARN extracts an item entity representation from the textual knowledge and contextual knowledge in the user-item KG. In the second step, KARN devises two key components SRA and A-SRA to extract the features of the user’s history interest and potential intent, respectively. Finally, based on the fusion of the representations of user’s history interest and potential intent, KARN calculates a probability to predict whether the user will click the candidate item.

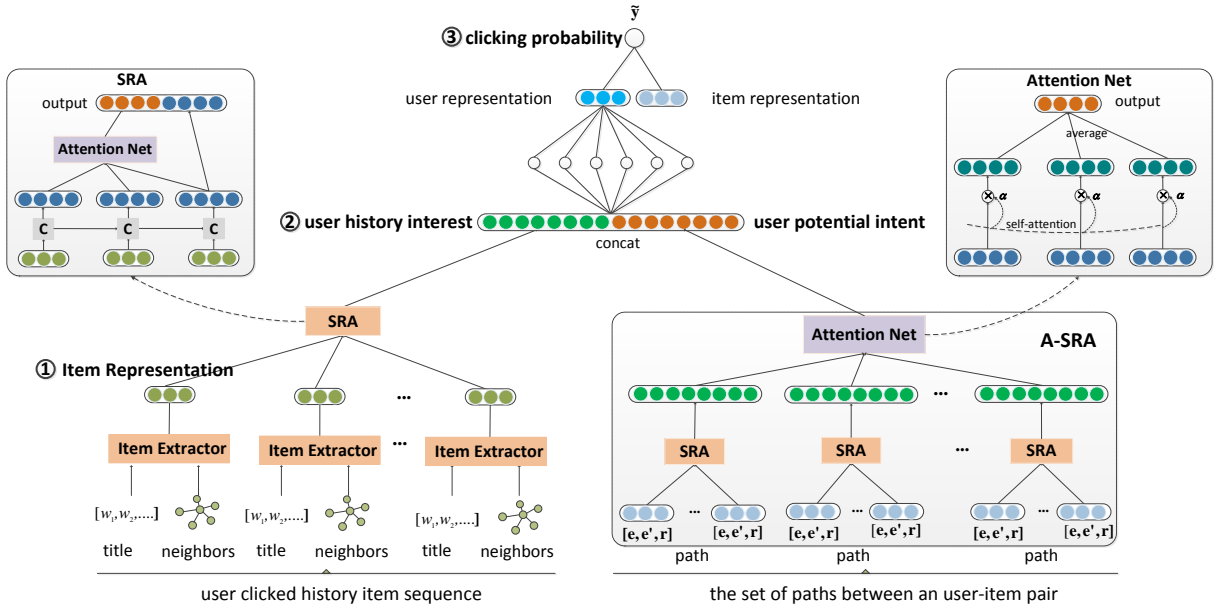


Figure 2: Simple visualization of KARN model.

### Item Entity Representation

Entities in the user-item knowledge graph usually include the user-entities, item-entities and attribution-entities (e.g., the category and brands of items). For an item entity such as book or movie in the KG, we extract item representation from the textual knowledge and contextual knowledge. Here we consider the title of the item as the textual knowledge due to the title usually gives the main topics of this item entity, and the immediate neighbors of items in the KG as the contextual knowledge since contextual entities are usually provide more complementary information in improving the identifiability of the item entity.

We denote the item entity as  $v$ , its textual knowledge as  $T = \{w_1, w_2, \dots\}$  and contextual knowledge as  $C = \{e_1, e_2, \dots\}$ . Accordingly, their embeddings are denoted as  $\mathbf{v} \in R^d$ ,  $\mathbf{T} = [\mathbf{w}_1, \mathbf{w}_2, \dots]$ ,  $\mathbf{w}_i \in R^{d'}$  and  $\mathbf{C} = [\mathbf{e}_1, \mathbf{e}_2, \dots]$ ,  $\mathbf{e}_i \in R^d$ , where  $d$  and  $d'$  are respectively the dimensions of embeddings of entities and words. For the textual knowledge of items, we use a convolution neural network composed of a convolutional layer and a local max-pooling layer to learn the textual representation of items. In the convolutional layer, we adopt the filter  $\mathbf{c}_i (i \in [1, \dots, F])$  to the input embedding  $\mathbf{T}$  and produce the feature maps  $\mathbf{m}_i = \text{ReLU}(\mathbf{c}_i \otimes \mathbf{T} + b_i)$ , where  $F$  is the number of filters,  $\otimes$  is the convolution operation and  $b_i$  is the bias term. In pooling layer, we use a local max-pooling operation on each feature map  $\mathbf{m}_i$ , and identify the most significant local features as  $\mathbf{p}_i$ . Then the concatenation of these local features is taken as the textual representation of item  $v$  as

$$\mathbf{v}' = g_1([\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_F])$$

where  $g_1$  is the transformation function that can be linear or non-linear and  $\mathbf{v}' \in R^d$ . For the contextual knowledge of items, the contextual representation is calculated as the

average of its contextual entities, i.e.,

$$\bar{\mathbf{v}} = \frac{\sum_{e_i \in C(v)} \mathbf{e}_i}{|C(v)|}$$

Based on the textual knowledge and contextual knowledge of items, we denote the item representation as

$$\mathbf{I} = g_2([\mathbf{v}, \mathbf{v}', \bar{\mathbf{v}}])$$

where  $\mathbf{I} \in R^d$  and  $g_2$  is the linear or non-linear transformation function.

### User's History Interest Extraction

With the goal of obtaining the user's history interest, SRA component is designed as the stack of the recurrent neural network and an attention network, which takes the user's clicked item sequence as input and outputs the user's history interest representation.

Based on the item entity representation extraction, we can get the representations of the input item sequence  $\{v_1, v_2, \dots, v_t\}$  as  $\{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_t\}$ . In  $j$ -step of LSTM, both the  $j$ -th item embedding  $\mathbf{I}_j$  and the output of  $j-1$ -step of LSTM are taken as inputs, and an intermediate representation  $\mathbf{h}_j \in R^m$  is outputted, which is given by:

$$\mathbf{h}_j = \text{LSTM}(\mathbf{h}_{j-1}, \mathbf{I}_j)$$

where  $m$  is the dimension of the hidden states of LSTM. Usually the last hidden state  $\mathbf{h}_t$  is regarded as the sequential features of the input sequence in prior works (Xiong, Hoang, and Wang 2017; Wang et al. 2019). Besides, we consider that the previous hidden states may make different contributions to generate the holistic sequential representation. Thus we take the hidden states  $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_t] \in R^{m \times t}$  of

LSTM as input, and adopt following attention mechanism to generate the user’s history interest representation  $\mathbf{s}$  as

$$\begin{aligned}\mathbf{A} &= \text{softmax}(\mathbf{W}_1 \sigma(\mathbf{W}_2 \mathbf{H})) \\ \mathbf{a} &= f_a(\mathbf{A} \mathbf{H}^T) \\ \mathbf{s} &= [\mathbf{a}; \mathbf{h}_t]\end{aligned}$$

where  $\sigma$  is the activation function, such as sigmoid, ReLu or tanh.  $\mathbf{W}_1 \in R^{s_2 \times s_1}$  and  $\mathbf{W}_2 \in R^{s_1 \times m}$  are weight matrices.  $s_2$  is designed for measuring how many different aspects of the input item sequence need to be focused on. Each row vector of the matrix  $\mathbf{A} \in R^{s_2 \times t}$  represents a weight distribution that weights the different contributions of the input items.  $f_a$  is the operator that calculates the average of the row vectors in matrix  $\mathbf{A} \mathbf{H}^T$ .

### User’s Potential Intent Extraction

Aiming to get the potential user’s intent from the path set between an user-item pair, A-SRA component is devised in two steps: 1) Extracting the single-path feature of each path in the path set, (2) Obtaining the user’s potential intent representation by integrating the multiple single-path features with different weights.

In the first step, given a path  $p_j = \{e_1, r_1, e_2, r_2, \dots, e_q\}$  with  $j = [1, \dots, n]$ , we concatenate the embeddings of the current entity  $e_i$ , its entity-type  $e'_i$  and relation  $r_i$  as the input vector  $\mathbf{x}_i = [\mathbf{e}_i; \mathbf{e}'_i; \mathbf{r}_i]$  of the cell of LSTM. Here we use the item representations for item entities, and define a null relation  $r_q$  for the last entity  $e_q$  to pad it on the end of the path. As such, the input path embedding is denoted as  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q\}$ , which not only contains the sequential information of the path, but also the compositional semantics information of the entity and its relation to the next entity.

Having established the input representation of the path  $p_j$  as  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q\}$ , we employ other SRA network to explore the sequential information of the path, and generate the single-path representation for encoding its holistic semantics as  $\mathbf{s}_j$ . For the path set  $P(u, v) = \{p_1, p_2, \dots, p_n\}$  between the user  $u$  and the item  $v$ , we can get its feature representations as  $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$ . As prior works (McCallum et al. 2017) studied, different paths have different impacts to model the potential user intent, thus we utilize other attention mechanism used in SRA component to produce the user’s potential intent representation  $\bar{\mathbf{s}}$ , which is the aggregation of the multiple single-path feature representations  $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$  with different weights.

### Probability

Based on the user’s history interest representation  $\mathbf{s}$  and the potential user’s intent representation  $\bar{\mathbf{s}}$ , we feed them into a project layer for obtaining the user representation as  $\mathbf{u} = \text{sigmoid}(\mathbf{U}_1[\mathbf{s}; \bar{\mathbf{s}}] + \mathbf{b}_1)$ . Finally, given the user representation  $\mathbf{u}$  and item representation  $\mathbf{I}$  of the item entity  $v$ , we calculate the probability of the user  $u$  clicking the item  $v$  as

$$\tilde{y} = \text{sigmoid}(\mathbf{U}_2^T[\mathbf{u}; \mathbf{I}] + b)$$

where  $\mathbf{U}_1 \in R^{d \times 4m}$ ,  $\mathbf{b}_1 \in R^d$ ,  $\mathbf{U}_2 \in R^{2d}$  and  $b$  are the parameters of the projection layers.

## Training

We use the observed facts of the user-item interactions as positive samples and unobserved facts as negative samples for training our model. A training sample is denoted as  $X = \{(u, v), S(u), P(u, v), y\}$ , where  $S(u) = \{v_1, v_2, \dots, v_t\}$  is the users’ history clicked item sequence,  $P(u, v) = \{p_1, p_2, \dots, p_n\}$  is the set of paths between user  $u$  and item  $v$ , and  $y$  is the label that equals to 1 if  $X$  is the positive sample, otherwise 0. After our model, each input sample has the respective estimated probabilities  $\tilde{y} \in [0, 1]$  of user  $u$  clicking item  $v$ . Thus we minimize the following negative log-likelihood function to train our model:

$$L = - \left\{ \sum_{X \in \Delta^+} y \log(\tilde{y}) + \sum_{X' \in \Delta^-} (1 - y) \log(1 - \tilde{y}) \right\}$$

where  $X$  and  $X'$  are respectively the positive and negative samples from the positive and negative set  $\Delta^+$  and  $\Delta^-$ .

**Construct Negative Samples.** We construct a negative training sample from three aspects based on a positive sample  $X = \{(u, v), S(u), P(u, v), 1\}$ : (1) For the user-item pair  $(u, v)$ , we randomly select the item  $v'$  that the user has not clicked to construct the negative user-item pair  $(u, v')$ , (2) For the user’s clicked sequence  $S(u) = \{v_1, v_2, \dots, v_t\}$ , we randomly replace  $o_1$  items in  $S(u)$  with items that the user has not clicked, where  $o_1 = \lceil (1 - \epsilon)t \rceil$ ,  $\lceil \cdot \rceil$  is the ceiling function and  $\epsilon \in [0, 1)$ , (3) For the path information  $P(u, v) = \{p_1, p_2, \dots, p_n\}$ , we respectively construct a negative path for each path in  $P(u, v)$ . Similar to the construction of the negative users’ clicked sequence, we also use  $o_2 = \lceil (1 - \epsilon)q \rceil$  entity-relation pairs  $\{(e'_i, r'_i) | i \in [1, o_2]\}$  selected from user-item KGs to randomly replace entity-relation pairs in the path  $p_j$ . Thus the negative training sample can be organized as  $X' = \{(u, v'), S'(u), P'(u, v), 0\}$ .

## Experiment

In this section, we evaluate our model KARN on Amazon<sup>1</sup> dataset. We first introduce the datasets and baselines, then present the corresponding experimental results and analysis.

### Experimental Setup

**DataSet** Amazon review dataset (McAuley et al. 2015; He and McAuley 2016) is an user-item interaction dataset that involves products in 24 categories, such as Books, Movies & TV and Music, etc. Each category contains three types of data, including user’s rating data, user’s review data and product’s meta-data (prices, brands). In this experiment, we use the 5-core<sup>2</sup> data of Books, Music, Movies & TV and Clothing as our datasets. Here we process these datasets in three steps: (1) We discard the users whose clicked items are fewer than 5 and sort the remain user’s clicked items by time-stamp for getting the users’ clicked sequences. (2) We use these three types of data to construct user-item knowledge graph according to the section ‘Problem Formulation’. (3) We generate the paths between the user-item pairs from

<sup>1</sup><http://jmcauley.ucsd.edu/data/amazon/>

<sup>2</sup>Each user or item has at least 5 associated reviews

Table 1: Statistics of the Datasets

DataSets		Book	Music	Movies &TV	Clothing
User-Item Interactions	#Users	8,026,324	478,235	2,088,620	3,117,268
	#Items	3,979,373	599,080	498,117	3,040,742
	#Interactions	31,405,196	900,712	6,304,580	6,027,597
	#Brands	67	346	2,814	8,495
	#Categories	3,986	478	801	2,802
Knowledge Graphs	#Entities	12,009,594	1,078,139	2,590,343	6,160,385
	#Triplets	95,430,838	8,652,526	12,956,338	81,140,545
Path	#Paths	34,769,400	662,321	6,878,390	4,149,625
	Avg Length	4.72	5.01	4.86	5.05

the user-item knowledge graph by vanilla BFS in two directions used in (Xiong, Hoang, and Wang 2017). For each category data, we randomly sampled 80%, 10% and 10% of the samples as the training, valid and test data .

**Parameter Setting** In training stage, we set the number of filters as  $F = 5$ , the maximum length of a path and clicked sequence as  $q = 8$  and  $t = 10$ , the maximum number of the path set as  $n = 6$ . We also apply a grid search to find out the best settings of hyper-parameters, i.e., we select the dimensions of entity embeddings  $d$ , word embeddings  $d'$  and hidden states  $m$  from  $\{50, 100, 200, 250, 300\}$ , the aspects of self-attention  $s_2$  from  $\{1, 3, 5, 7, 9\}$ , the batch size  $B$  from  $\{500, 1000, 1500, 2000\}$ , and the learning rate  $\eta$  from  $\{0.1, 0.01, 0.001, 0.0001\}$ . We train our model 1000 epochs and the best optimal parameter configurations are  $d = d' = m = 100$ ,  $s_2 = 5$ ,  $B = 1000$ ,  $\eta = 0.001$ . To compare with baselines, we use Hit@ $K$  and ndcg@ $K$  with  $K = \{1, 2, \dots, 10\}$  as the evaluation metrics.

## Baselines

We use the following state-of-the-art recommendation methods as baselines in our experiments.

- BPR (Rendle et al. 2009) is the bayesian ranking model that uses Matrix Factorization (MF) as the prediction component for recommendation. It utilizes the user-item interaction to learn representations of users and items.
- NFM (He and Chua 2017) combines the linearity of Factorization Machines (FM) and the non-linearity of neural network for performing item recommendation. It treats historical items as the features of users.
- CKE (Zhang et al. 2016) employs MF and TransR (Lin et al. 2015) to integrate text, images, and structure information of knowledge graph for recommendation. It is the KG-enhanced method that utilizes KG to guide the representation learning of items.
- FMG (Zhao et al. 2017) is a meta-path based model that employs FM to assemble user or item vectors factorized by meta-path similarity matrices using MF for rating recommendation.
- DAN (Zhu et al. 2019) is a deep attention neural network that extracts user features from the user’s history clicked sequence for performing recommendation.
- KPRN (Wang et al. 2019) is a KG-aware path recurrent network that reasons on paths to infer the underlying rationale of an user-item interaction.

## Results and Analysis

In this experiment, we compare our model with several state-of-the-art baselines on top- $K$  recommendation. For the compared baselines, we rerun them using our datasets. Further we give the convinced experimental results in Figure 3.

From Figure 3, we can observe that our model KARN significantly outperforms all baselines on four datasets with all metrics. Specifically, KARN achieves at least 2.5% on Hits@ $K$  and 1.8% on ndcg@ $K$  higher performance than other models. The three advantages of KARN make its superiority of recommendation performance: (1) We incorporate the user’s history clicked sequence and path connectivity between the user-item pairs for personalized recommendation. (2) We develop an attentional recurrent neural network SRA to learn the holistic sequential representations of the input sequences. (3) We employ attention mechanism in A-SRA component to learn the comprehensive path representation, which is integrated from the different paths between the user-item pairs with a combination of different weights.

More detailed analysis is as follows: (1) Among all baselines, the meta-path method FMG that heavily relies on the predefined meta-paths, has the poor performance on four datasets. It is mainly because that the meta-path patterns may fail to fully explore the user-item connectivity when introducing remote entities. (2) The models with rich auxiliary information (DAN, CKE and NFM) perform better in general than the shallow methods (BPR), which suggests that the introduced KG and user’s history clicked items are capable of learning user-item interaction information for recommendation effectively. (3) For the neural network based models DAN and NFM, DAN has the better performance in our experiments. The reason is that the attention mechanism used in DAN can better capture the sequential features of the user’s clicked sequences than NFM. (4) Expect for our model, KPRN substantially achieves higher performance than FMG among the path-based models. It indicates that KPRN can well explore the user-item connectivity by reasoning on paths to infer the user preferences.

## Discussion on Different KARN Variants

To investigate the efficacy of the design of our model, we explore the variants of our model with respective to following aspects: (1) the usage of user’s clicked history sequence, (2) the usage of path connectivity in user-item knowledge graph, (3) the usage of textual knowledge, (4) the usage of contextual knowledge, and (5) the usage of attention mechanism. For convenience, we denote KARN-s and KARN-p as the variants that respectively does not consider the user’s clicked history sequence and path connectivity information for recommendation. Moreover, KARN-t and KARN-c are the variants that either textual knowledge or contextual knowledge is not used for extracting item representations, and the variant KARN-a is to exclude the attention mechanism from our model KARN.

The experimental results of these variants are reported on Table 2. From this table, we can observe that: (1) Both KARN-s and KARN-p obtain lower recommendation performance than KARN. It proves that the user’s clicked history sequence and path connectivity between user-item pairs

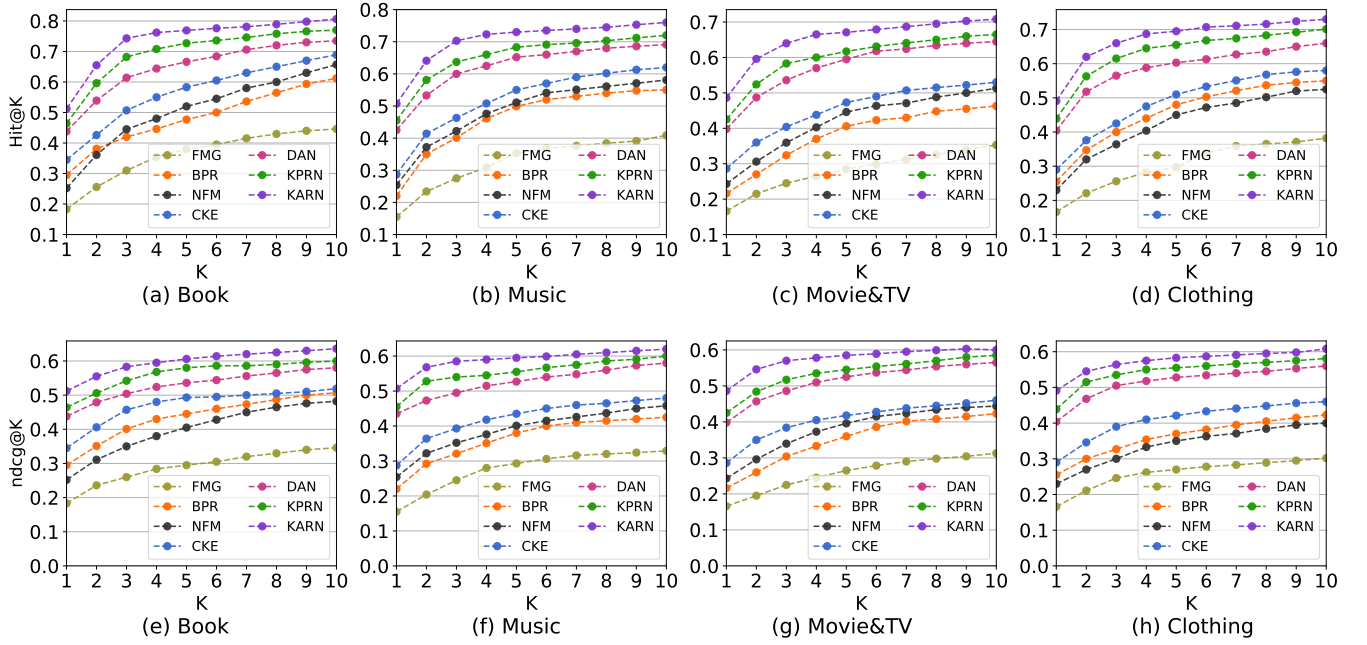


Figure 3: Recommendation Performance of KARN on Hit@K and ndcg@K

in KGs are important factors to infer the user’s preferences towards target items. That is, both factors make significant contributions on improving the personalized recommendation performance. (2) There is the gap recommendation performance between KARN and KARN-t (KARN-c), which indicates that KARN successfully alleviates textual knowledge and contextual knowledge to learn item representations. It is because that such two aspects of knowledge can provide the indispensable semantic characteristics of items. (3) KARN has the higher performance than KARN-a on all datasets with all metrics. The main reason is that the attention mechanism in KARN can dynamically learn the holistic sequence representations of the input sequences, in forms of either the user’s clicked history sequence or the path sequence, with assigning different weights to different items.

### Effect of Different Multi-Aspects $s_2$

The above experiments show that the usage of attention mechanism in our model is beneficial for improving recommendation performance. The attention mechanism is a multi-head attention, where the parameter  $s_2$  measures how many different aspects of the input sequences need to be concerned on. This subsection mainly evaluates how the performance of our model KARN changes with different parameter  $s_2$ . In this experiment, we select  $s_2$  from  $\{1, 2, \dots, 10\}$  and use the optimal configurations for the rest parameters. Figure 4 gives the convinced results on Hit@10 and ndcg@10, which demonstrates that: (1) Our model KARN achieves highest performance when  $s_2 = 5$  on both Hit@10 and ndcg@10. It suggests that  $s_2 = 5$  can best capture the holistic semantic features from the multiple aspects of the input sequences. (2) As a whole, the accuracy of our model

first increases and then decreases with  $s_2$  increasing. That is, too small  $s_2$  can hardly explore the integration relatedness of multiple aspects of the inputs, and too large  $s_2$  may bring much more noises than useful signals.

### Related Works

With the explosive growth of information, it is increasingly important to make personalized recommendation for satisfying user’s preferences. A variety of previous works, such as CF-based methods (Das et al. 2007; Xue et al. 2017), Content based methods (IJntema et al. 2010; Zhu et al. 2019) and hybrid based methods (Li et al. 2011; Morales, Gionis, and Lucchese 2012), have been done for trying to tackle personalized recommendation. These methods usually use various unstructured information, such as textual reviews, visual images, and various implicit or explicit feedbacks for making personalized recommendation, which have weakness in the explainable recommendation.

Recently, many works that attempt to use knowledge base embedding for explainable recommendation, have been done. Such knowledge-aware efforts can be roughly categorized into embedding-based and path-based methods. Embedding-based methods (Wang, Wang, and Yeung 2015; He, Kang, and McAuley 2017; Huang et al. 2018; Wang et al. 2018b; Zhang et al. 2016; Dong, Chawla, and Swami 2017; Velickovic et al. 2018) usually leverage knowledge graph embedding techniques (Bordes et al. 2013; Lin et al. 2015; Zhou et al. 2017) to guide the representation learning of items and users. For example, CKE (Zhang et al. 2016) utilizes embedding model TransR (Lin et al. 2015) to embed items into a ‘transition space’, where users are modeled as translation vectors operating on items. Such meth-

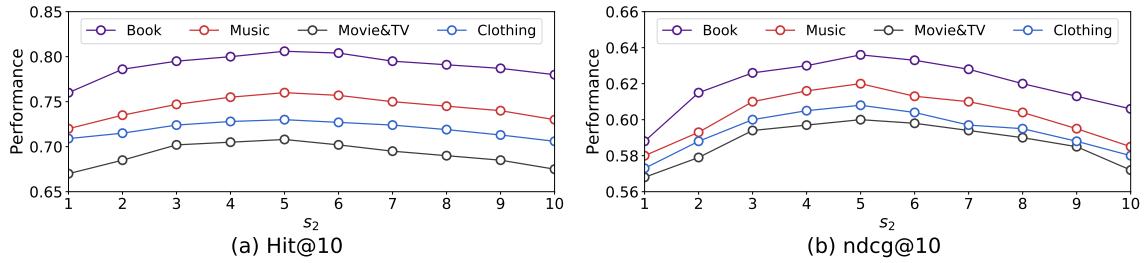


Table 2: Performance of Different Variants of KARN on Hit@K and ndcg@K with  $K = \{1, 5, 10\}$ 

Variants	Book			Music			Movies &TV			Clothing		
	Hit@1	5	10	Hit@1	5	10	Hit@1	5	10	Hit@1	5	10
KARN-s	49.6	75.4	78.8	48.9	71.7	74.7	46.3	65.7	68.6	47.6	67.6	72.3
KARN-p	49.3	75.2	77.3	48.4	71.4	74.5	46.6	65.2	68.1	47.0	68.1	72.0
KARN-t	50.0	76.1	79.5	49.3	72.3	75.1	47.0	65.0	69.3	47.9	66.6	72.4
KARN-c	50.2	75.6	79.8	49.2	72.0	74.8	47.4	64.8	69.0	47.4	67.2	72.9
KARN-a	49.0	74.3	78.2	48.1	70.9	74.0	46.8	65.2	68.7	47.3	67.8	72.7
KARN	<b>51.2</b>	<b>76.9</b>	<b>80.6</b>	<b>50.7</b>	<b>73.0</b>	<b>76.0</b>	<b>48.6</b>	<b>67.1</b>	<b>70.8</b>	<b>49.1</b>	<b>69.5</b>	<b>73.6</b>

Variants	ndcg@1			ndcg@1			ndcg@1			ndcg@1		
	ndcg@1	5	10	ndcg@1	5	10	ndcg@1	5	10	ndcg@1	5	10
KARN-s	49.6	59.1	61.8	48.9	57.1	59.4	46.3	57.1	58.9	47.6	57.1	58.6
KARN-p	49.3	58.5	60.9	48.4	56.6	58.8	46.6	56.3	58.0	47.0	56.5	58.0
KARN-t	50.0	59.8	63.1	49.3	57.8	60.2	47.0	58.1	59.5	47.9	57.6	59.2
KARN-c	50.2	59.4	62.6	49.2	57.4	59.5	47.4	57.4	59.7	47.4	57.4	58.9
KARN-a	49.0	59.0	62.0	48.1	56.8	58.7	46.8	56.5	59.0	47.3	56.8	57.7
KARN	<b>51.2</b>	<b>60.6</b>	<b>63.6</b>	<b>50.7</b>	<b>58.5</b>	<b>61.0</b>	<b>48.6</b>	<b>58.8</b>	<b>60.8</b>	<b>49.1</b>	<b>58.3</b>	<b>60.0</b>

Figure 4: The Performance of KARN on Different Parameter  $s_2$ .

ods explore one-step connectivity between users and items in KGs as the guidance of representation learning. However, this regularization of knowledge graph embedding has not fully exploit the connectivity characterization between users and items, since it lacks the reasoning ability to explain why an item is recommended for an user.

For addressing above issues, path-based methods (Wang et al. 2018a; Zhao et al. 2017; Heitmann and Hayes 2010; Sun and Han 2012; Hu et al. 2018; Sun et al. 2018; Wang et al. 2019) introduce the path connectivity information in KGs to perform the recommendation learning. FMG (Zhao et al. 2017) predefines meta-path patterns to capture the user-item affinities carried in KGs for learning the representations of the users and items, where meta-paths are a series of sequences of entity type generated from amount prior domain knowledge. These meta-path-aware models have limited ability of generalization on the unseen interactions because they heavily rely on the quality of meta-paths. Recently, many studies attempt to learn the representations of the paths between users and items for inferring users' preferences. For instance, RKGE (Sun et al. 2018) and MCRec (Hu et al. 2018) respectively employ RNN and CNN over the input embeddings of entities along a path to get the feature representation of the path between user-item pairs for recommendation. KPRN (Wang et al. 2019) considers relations, entity and entity-type together along paths, and learns path representations using an attentional RNN to integrate multiple paths between an user-item pair with different weights.

KPRN is the current state-of-the-art path-based recommendation model. However, such works only explore path information in KGs, but ignore user's history clicked sequential behaviors for personalized recommendation. The user's history clicked sequential behaviors is a better reflection of the user's recent interests. Towards this end, we propose a knowledge-aware attentional reasoning network that incorporates the user's clicked history sequence and path connectivity information in KGs for more precise recommendation.

## Conclusion

In this paper, we propose a knowledge-aware attentional reasoning network KARN for personalized recommendation, which considers not only the user's history clicked sequence but also the path connectivity between users and items in KGs. For the user's history clicked sequence, KARN develops the SRA component composed of the recurrent neural network and attention network to extract the user's history interest. For the path information in KGs, KARN devises the A-SRA component that is a hierarchical attention network to capture the potential user's intent. By leveraging the fusion of the user's history interest and potential intent, KARN gives the clicking probability of whether the user clicks the candidate item. We empirically conduct evaluation experiments on Amazon data. The experimental results demonstrate that our model KARN has achieved state-of-the-art recommendation performance.

## Acknowledgments

This work is supported by National Key R&D Program No.2017YFB0803003, and the National Natural Science Foundation of China (No.61202226). We thank all anonymous reviewers for their constructive comments.

## References

- Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*, 2787–2795.
- Cao, Y.; Hou, L.; Li, J.; and Liu, Z. 2018. Neural collective entity linking. In *COLING*, 675–686.
- Catherine, R., and Cohen, W. W. 2016. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *The Conference on Recommender Systems*, 325–332.
- Das, A.; Datar, M.; Garg, A.; and Rajaram, S. 2007. Google news personalization: scalable online collaborative filtering. In *WWW*, 271–280.
- Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *SIGKDD*, 135–144.
- He, X., and Chua, T. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*, 355–364.
- He, R., and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, 507–517.
- He, S.; Liu, K.; Zhang, Y.; Xu, L.; and Zhao, J. 2014. Question answering over linked data using first-order logic. In *EMNLP*, 1092–1103.
- He, R.; Kang, W.; and McAuley, J. 2017. Translation-based recommendation. In *RecSys*, 161–169.
- Heitmann, B., and Hayes, C. 2010. Using linked data to build open, collaborative recommender systems. In *AAAI*.
- Hu, B.; Shi, C.; Zhao, W. X.; and Yu, P. S. 2018. Leveraging meta-path based context for top-*N* recommendation with a neural co-attention model. In *SIGKDD*, 1531–1540.
- Huang, J.; Zhao, W. X.; Dou, H.; Wen, J.; and Chang, E. Y. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *SIGIR*, 505–514.
- Intema, W.; Goossen, F.; Frasincar, F.; and Hogenboom, F. 2010. Ontology-based news recommendation. In *EDBT/ICDT*.
- Li, L.; Wang, D.; Li, T.; Knox, D.; and Padmanabhan, B. 2011. SCENE: a scalable two-stage personalized news recommendation system. In *SIGIR*, 125–134.
- Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 15, 2181–2187.
- McAuley, J. J.; Targett, C.; Shi, Q.; and van den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*, 43–52.
- McCallum, A.; Neelakantan, A.; Das, R.; and Belanger, D. 2017. Chains of reasoning over entities, relations, and text using recurrent neural networks. In *EACL*, 132–141.
- Morales, G. D. F.; Gionis, A.; and Lucchese, C. 2012. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *WSDM*, 153–162.
- Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: bayesian personalized ranking from implicit feedback. In *UAI*, 452–461.
- Sun, Y., and Han, J. 2012. Mining heterogeneous information networks: a structural analysis approach. *SIGKDD* 14(2):20–28.
- Sun, Z.; Yang, J.; Zhang, J.; Bozzon, A.; Huang, L.; and Xu, C. 2018. Recurrent knowledge graph embedding for effective recommendation. In *RecSys*, 297–305.
- Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.
- Wang, H.; Zhang, F.; Wang, J.; Zhao, M.; Li, W.; Xie, X.; and Guo, M. 2018a. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *CIKM*, 417–426.
- Wang, H.; Zhang, F.; Xie, X.; and Guo, M. 2018b. DKN: deep knowledge-aware network for news recommendation. In *WWW*, 1835–1844.
- Wang, X.; Wang, D.; Xu, C.; He, X.; Cao, Y.; and Chua, T. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*.
- Wang, H.; Wang, N.; and Yeung, D. 2015. Collaborative deep learning for recommender systems. In *SIGKDD*, 1235–1244.
- Xiong, W.; Hoang, T.; and Wang, W. Y. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *EMNLP*, 564–573.
- Xue, H.; Dai, X.; Zhang, J.; Huang, S.; and Chen, J. 2017. Deep matrix factorization models for recommender systems. In *IJCAI*, 3203–3209.
- Zhang, F.; Yuan, N. J.; Lian, D.; Xie, X.; and Ma, W. 2016. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*, 353–362.
- Zhang, Y.; Ai, Q.; Chen, X.; and Wang, P. 2018. Learning over knowledge-base embeddings for recommendation.
- Zhang, L.; Liu, P.; and Gulla, J. A. 2018. A deep joint network for session-based news recommendations with contextual augmentation. In *Hypertext and Social Media*, 201–209.
- Zhao, H.; Yao, Q.; Li, J.; Song, Y.; and Lee, D. L. 2017. Meta-graph based recommendation fusion over heterogeneous information networks. In *SIGKDD*, 635–644.
- Zhou, X.; Zhu, Q.; Liu, P.; and Guo, L. 2017. Learning knowledge embeddings by combining limit-based scoring loss. In *CIKM*, 1009–1018.
- Zhu, Q.; Zhou, X.; Song, Z.; Tan, J.; and Guo, L. 2019. DAN: deep attention neural network for news recommendation. In *AAAI*.