IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 29, 2020, accepted July 20, 2020, date of publication July 22, 2020, date of current version July 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3011279

MRP2Rec: Exploring Multiple-Step Relation Path Semantics for Knowledge Graph-Based Recommendations

TING WANG[1], DAQIAN SHI[2], ZHAODAN WANG[3], SHUAI XU[4], (Member, IEEE), AND HAO XU[1]

[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]ICT International Doctoral School, University of Trento, 38123 Trento, Italy
[3]College of Aviation Foundation, Aviation University of Air Force, Changchun 130012, China
[4]College of Electronic Science and Engineering, Jilin University, Changchun 130012, China

Corresponding author: Hao Xu (xuhao@jlu.edu.cn)

This work was supported in part by the Development Project of Jilin Province of China under Grant 20170203002GX and Grant 20170101006JC, in part by the Ministry of Science and Technology of the People's Republic of China under Grant 2018YFC2002500, in part by the Jilin Province Development and Reform Commission, China, under Grant 2019C053-1, in part by the Education Department of Jilin Province, China, under Grant JJKH20200993K, and in part by the Department of Science and Technology of Jilin Province, China, under Grant 20200801002GH.

ABSTRACT Knowledge graphs (KGs) have been proven to be effective for improving the performance of recommender systems. KGs can store rich side information and relieve the data sparsity problem. There are many linked attributes between entity pairs (e.g., items and users) in KGs, which can be called multiple-step relation paths. Existing methods do not sufficiently exploit the information encoded in KGs. In this paper, we propose MRP2Rec to explore various semantic relations in multiple-step relation paths to improve recommendation performance. The knowledge representation learning approach is used in our method to learn and represent multiple-step relation paths, and they are further utilized to generate prediction lists by inner products in top-K recommendations. Experiments on two real-world datasets demonstrate that our model achieves higher performance compared with many state-of-the-art baselines.

INDEX TERMS Recommender systems, knowledge graph, semantic representation.

I. INTRODUCTION

Recommender systems (RS) have become increasingly important for presenting information to users that meets their personalized preferences. However, it usually suffers from the data sparsity problem. To improve the performance of RS, researchers have proposed utilizing side information [1], [2] to enrich data. Traditional methods [3], [4] use supervised learning methods for encoding side information into item representations to predict user behaviours. Knowledge graphs (KGs), as an important side information source, have shown great potential on RS since KGs can link various types of information related to items into a unified global space. According to the idea that paths connecting entity pairs have various latent semantic features that can be used to represent entities, exploring paths in KGs can promote user preference prediction. Some methods, such as meta-path-based models [5], [6], mainly utilize path-based semantic relatedness coded in KGs as prediction relevance. However, these methods rely on manually designed features, thus hindering the improvement of recommendation performance since it is difficult to design optimal paths that capture unlimited and unaggregated knowledge.

The development of representation learning (RL) has led to studies that exploit knowledge graph embedding methods to represent instances with path semantic features in KGs for recommendations, and some works [7]–[9] have achieved good performance. However, the deficiency of these methods is that they do not consider the different semantic relations between each data instance encoded in multistep relation paths. Therefore, they cannot fully distil the latent features of both item similarity and collaborative user behaviour, which limits improving the recommendation performance.

The associate editor coordinating the review of this manuscript and approving it for publication was Wajahat Ali Khan.

To solve the problems mentioned above, in this paper, we propose a novel recommendation framework named MRP2Rec, which can encode the high-order semantic relations between entity pairs from KGs into user/item embeddings for recommendations without handcrafted features. More specifically, we first construct a unified graph for transferring independent instances into a unified global space. Then, we leverage the knowledge representation learning method to obtain sophisticated multiple-step relation path embedding for further generating prediction lists by the inner product in top-K recommendations. Our model is theoretically superior to existing methods because various semantic relations can be exploited from multiple-step relation paths. It can recommend items to users with implicit features of user collaborative behaviour as well as item similarity. Three major contributions of this paper are generalized as follows:

- We propose a novel knowledge graph-based recommendation framework, MRP2Rec, to perform recommendations with latent features of item similarity and collaborative user behaviour exploited from multiple-step relation paths.
- We develop a modified knowledge representation learning method to extract various semantic relations from a constructed unified knowledge graph for a recommendation.
- Experiments on two datasets demonstrate that our model achieves better performance compared with many state-of-the-art baselines.

The rest of this paper is organized into four sections. Section II introduces the relevant research of this paper. Section III details the methodology for constructing the MRP2Rec model, and Section IV discusses the experimental results to evaluate our model. Section V presents the conclusion and future work of this research.

II. RELATED WORK

To improve recommendation performance with side information, Rendle [3] proposed a feature-based factorization model LibFM to transform latent features, and Cheng *et al.* [4] proposed Wide&Deep to combine a (deep) nonlinear channel with a (wide) linear channel to learn valuable knowledge. Recently, an increasing number of studies have focused on utilizing KGs storing side information to improve recommendation performance. For example, Catherine and Cohen [10] combined the strengths of latent factorization with graphs in their model to make knowledge graph-based recommendations. In the studies of knowledge graph-based recommender systems, an important problem is how to fully exploit path semantics encoded in KGs. Existing models on this issue can be summarized by two methods: path-based methods and embedding-based methods.

A. PATH-BASED METHODS

Meta-path, connecting two instances with a sequence of relations, is popularly used for capturing relevant semantic features carried by KGs for recommendations. Several studies directly utilize path-based semantic relatedness coded in KGs as recommendation relevance. For instance, personalized entity recommendation (PER) [5] and Hete-CF [11] combine each user's multiple types of relations through manually defined meta-paths for a high-quality recommendation. These methods do not concern the relation semantics of each instance coded in meta-paths; thus, they perform unsatisfactorily on datasets with multiple-relation data. For capturing semantic relations, Shi *et al.* [6] and Daqian *et al.* [12] proposed to infer user preferences over path semantics depicted through weighted meta-path concepts derived from different link attribute values. Hu *et al.* [13] further proposed a novel DNN module with a co-attention mechanism MCRec to combine user embeddings and item embeddings with path-based context embeddings that make use of the mutual effect between path semantics and user behaviours.

To better integrate knowledge graphs into recommender systems, some works learn transformed features from path-based similarities of KGs to enhance original user or item representations for recommendations. For example, the matrix factorization-based dual regularization framework SimMF [2] mainly extends LFM [1] with entity (e.g., item) similarities derived from meta-path-based latent features of KGs where entity relations of different semantics are represented. Similarly, Zhao *et al.* [14] generated transformed features for instances from meta-path-based similarities through a "matrix factorization (MF) + factorization machine (FM)" approach. Although these matrix factorization methods are effective for extracting path-based latent features for recommendations, the main shortcoming is that they have limited scalability. Moreover, all the above meta-path-based methods do not perform well on various nodes and multiple relations data because they heavily rely on manually designed features to represent path semantics, which requires domain knowledge. For these reasons, the improvement of the recommendation performance is limited to some degree.

B. EMBEDDING-BASED METHODS

Knowledge graph embedding-based methods are widely used, inspired by representation learning (RL), where heterogeneous knowledge entities and relations are projected into a unified embedding space. Compared with path-based methods, embedding-based methods can automatically capture rich information from multitype item properties and user behaviours. Many prior efforts have utilized KG embedding-based methods in recommender systems. For example, Zhang *et al.* [15] proposed a collaborative knowledge base embedding (CKE) model that extends the collaborative filtering (CF) module [16] by introducing knowledge, image and text embedding of items. Wang *et al.* [17] constructed a novel neural network model called DKN for learning text embeddings combined with entity embeddings in news recommendation scenarios. Wang *et al.* [18] designed SHINE embedding sentiment networks, social networks, and profile networks with deep autoencoders for recommendations. These methods unify various types of side information in the recommender system but do not sufficiently learn the knowledge-level embeddings of entities, which is powerful for recommendation systems. For integrating large-scale heterogeneous data, collaborative filtering with knowledge graph (CFKG) [19] extends CF by proposing a representation learning approach that embeds heterogeneous entities for a personalized recommendation. CoFM [20] combines the recommendation task and knowledge graph learning task by sharing parameters of aligned items and entities. HI2Rec [21] and ADPE [22] further exploit the semantics of both entities and relations from KGs based on a knowledge representation learning approach for the recommendation. Considering the incomplete nature of KG, Wang *et al.* [23] and Cao *et al.* [24] proposed to jointly learn the task of recommendation and knowledge graph completion. Xin [25] proposed the RCF model to describe items with both relation type embeddings and relation value embeddings. Zhao *et al.* [26] exploited rich user-related behaviours in the graph by their proposed model IntentGC. Although these models have achieved significant improvement, they do not consider the complex high-order semantic relations of entity pairs, which has already been widely researched in path-based methods and largely confines the improvement of these methods' performance.

Recent state-of-the-art studies have attempted to encode high-order path semantics into user/item embeddings so that they can fully exploit user preferences from KGs that have various nodes and multiple relations. Sha *et al.* [27] proposed AKGE to learn user and candidate item representations by propagating information in a subgraph of this user-item pair. RippleNet [28] exploits user preferences through preference propagation that can automatically discover possible paths from an item that is observed as having been interacted with by a user to a candidate item. However, the importance of relations cannot be seriously characterized, since the embedding matrix of a relation R is difficult to train to exploit the sense of significance in the quadratic form $v^T Rh$. Wang *et al.* [29] proposed KGCN to predict the representation of item v_i by aggregating the embedding of entities according to distance neighbours in KGs. A follow-up method, KGCN-LS [30], extends KGCN with a label smoothness (LS) mechanism to obtain a comprehensive representation of item v_i. Similarly, Wang *et al.* [8] proposed a graph neural network that refines the node's (e.g., items) embeddings by propagating recursively from its neighbours for encoding high-order relations. For unified refining user embeddings and item embeddings, Qu *et al.* [31] proposed KNI to consider the interaction between item-side neighbours and user-side neighbours. However, these methods ignore the different semantic relations under paths between paired entities. Recurrent knowledge graph embedding (RKGE) [7] and knowledge-aware path recurrent networks (KPRNs) [9] utilize recurrent networks that leverage the information of semantic paths between entity pairs to characterize user preferences towards items.

These methods characterize the importance of various semantic relations, but they only consider paths that start from a user and end with an item so that they fail to embed valuable path semantics adequately.

In contrast to the current state-of-the-art models, our MRP2Rec model makes it possible to fully automatically learn different semantic relations. By modifying the knowledge representation learning approach, our model can fully exploit significant semantic relations from paths encoded in KGs. The advantage of MRP2Rec is that latent features of user collaborative behaviour, as well as item similarity, can be learned from KGs to enhance recommendation performance. Seven baselines were chosen to verify the superiority of our model. These baselines cover almost all kinds of methodologies in the knowledge graph-based recommendation area, including the traditional supervised learning method [3], [4], path-based method [5], and embedding-based method [15], [17], [20], [28].

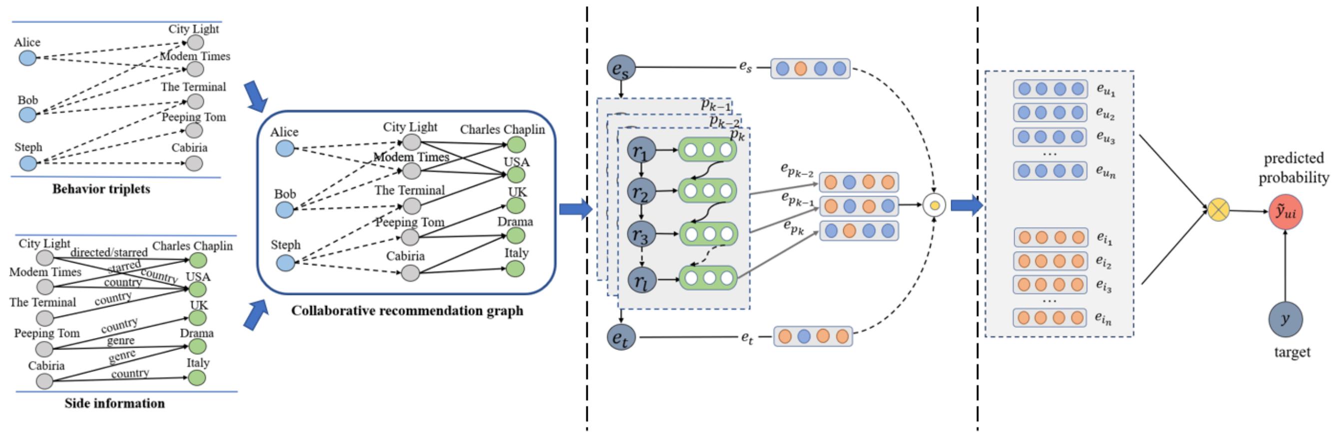
III. PROPOSED METHODOLOGY

In a recommendation scenario, historical user-item interactions indicate that user preferences for items are typically used. In addition, side information as auxiliary data stored in knowledge graphs that are composed of real-world entities and relationships can help to alleviate the sparsity of user-item interactions. Inspired by [32], different semantic relations of paths are expected to be captured from KGs, and high-quality user/item representation can be learned for accurate user profiling and personalized recommendation. Following this idea, we propose a novel knowledge graph-based recommendation framework, MRP2Rec, and an overview of our model is shown in Fig. 1. We first obtain paths from a collaborative recommendation graph structure that was defined to model user-item interactions and auxiliary data into a unified global space. Then, we employ a neural network structure that translates the relation path embeddings and makes the embeddings of head entities close to the embeddings of tail entities so that user and item embeddings with path semantics can be learned. Finally, we utilize these knowledge embeddings to predict candidate items that may be clicked by users. For clarity, some important symbols used throughout this paper are listed in Table 1.

A. MULTIPLE-RELATION PATHS MINING

An intuitive method for learning side information from a KG is to learn item attributes while preserving the KG structure and then integrating them with latent features learned from user behaviours. Heterogeneous data prevent full path semantic mining since latent features from user behaviours and side information are not visible for each other. To address such heterogeneous data in a unified space, we define a collaborative recommendation graph (CRG) structure specialized for the recommendation that can help to embed multiple-step relation paths into a low-dimensional space, and the details are described below.

We use graph G_1, which is defined as $\{(u, y_{ui}, i)|u \in U, i \in I\}$, where U denotes the user set and I denotes the item set, to store historical user behaviour triplets. y_{ui} is a



(a) Multiple-relation paths mining (b) Path semantic embedding (c) Personalized recommendation

FIGURE 1. The framework of the MRP2Rec model. Its inputs are the historical user-item interactions (behaviour triplets) and side information. (a) is the data processing module where a collaborative recommendation graph was constructed for path mining. (b) is the embedding layer in which multiple-relation paths were exploited to represent users and items. (c) is the prediction layer where user and item representations were used to compute the prediction list in a top-K recommendation scenario. The outputs are the predicted probability of the user to interact with the candidate item.

TABLE 1. A summary of our symbols and descriptions.

Symbol	Description
u	The representation of user u
i	The representation of item i
$\hat{y}_{ui}$	The predicted probability that user u interacts with the candidate item i
h	The representation of a head entity
r	The relation between entity pairs
t	The representation of a tail entity
e	e_s is the feature vector of source entity in a multiple-step relation path, e_t is the feature vector of a target entity in a multiple-step relation path, e_r is the feature vector of the relation between an entity pair, e_u and e_i is the feature vector of user u and item i
p	The representation of a relation path
P	The set of a relation path
π	The representation of a multiple-step relation path
Π	The set of paths
E	The set of entities

constant that can only be 0 or 1, where $y_{ui} = 1$ indicates user u and item i have the relationship of interacting; otherwise, $y_{ui} = 0$. Graph $G_2 = \{(h, r, t) \mid h, t \in E, r \in R\}$ utilizes triplets to describe the relationship r from head entity h to tail entity t for organizing the side information. Moreover, $A = \{(i, e) \mid i \in I, e \in E\}$ is established to construct the CRG $G = \left\{(h, r, t) \mid h, t \in E', r \in R'\right\}$, where $E' = E \cup U$ and $R' = R \cup y_{ui}$. Moreover, we manually add triplets with reverse relationships to enrich the semantics of CRG.

For mining paths from given triplets in CRG, we first perform random walks from the source entity e_s to the target entity e_t following relations connecting them. Multiple-step relation paths and relation paths $p = \{r_1, r_2, \ldots, r_l\} \in P$ containing a sequence of relations are obtained. We can write the relation path as $E_0 \xrightarrow{r_1} E_1 \xrightarrow{r_2} \ldots \xrightarrow{r_l} E_l$, where $E_0 = e_s$ and $e_t \in E_l$. For any entity $e_m \in E_i$, the random walk probability from target entity e_s following exact relations in p can be computed as:

$$R_p(m) = \sum\nolimits_{n \in E_{i-1}(\cdot, m)} \frac{1}{|E_i(n, \cdot)|} R_p(n) \tag{1}$$

where $E_i(n, \cdot)$ is the direct successor of $n \in E_{i-1}$ following the relation r_i and $E_{i-1}(\cdot, m)$ is the direct predecessor along relation r_i in E_{i-1}. For target entity e_s in each relation path p, we set the random walk probability to equal one. By performing recursively from e_s through p, the reliability of the path from target entity e_s to tail entity e_t is:

$$R(p|s, t) = R_p(t). \tag{2}$$

We further utilize the probability for each relation path to suggest which paths should be given more attention when there are multiple paths between the same entity pairs.

We obtain complete paths π by expanding all relation paths p between the same entities (e_s, e_t) such that $\pi = \left\{e_s, p_1, \ldots, p_k, e_t\right\} \in \Pi$. These paths are further processed by the embedding layer of the MRP2Rec model to automatically learn user and item representations for the recommendation.

B. PATH SEMANTICS EMBEDDING

Formally, for a triplet (s, p, t), assume the embeddings of s, p, t are e_s, e_p, e_t, respectively, then we want the representation of e_t to contain multiple path semantics. As explained in the above section, the number of k multiple-step relation paths between two entities (e_s, e_t) is defined as

$\pi = \{e_s, p_1, \ldots, p_k, e_t\} \in \Pi$, and $e_s \in R^d$ and $e_t \in R^d$ are the embeddings of the target entity and tail entity. Each relation path p_j in path π is $p_j = \{r_1, r_2, \ldots, r_l\}$, $j \in \{0, 1, \ldots, k\}$, and each relation $r_i, i \in \{0, 1, \ldots, l\}$ is represented as an embedding vector $r_i \in R^d$. We use the LSTM module [33] to learn the representations of the relation terms r in p from left to right, as illustrated in Fig. 2. At the path step $i-1$, the LSTM module outputs a hidden state vector e_{i-1}, consuming the subsequence $\{r_1, r_2, \ldots, r_{i-1}\}$. Simultaneously, the embedding of the relation r_{i-1} is the input vector at this step. Consequently, e_{i-1} and r_{i-1} are utilized to learn the hidden state of the next step i:

$$\begin{aligned} \tilde{c}_i &= \tanh\left(W_c\left[e_{i-1}; r_i\right] + b_c\right) \\ f_i &= \sigma(W_f\left[e_{i-1}; r_i\right] + b_f) \\ u_i &= \sigma(W_u\left[e_{i-1}; r_i\right] + b_u) \\ o_i &= \sigma\left(W_o\left[e_{i-1}; r_i\right] + b_o\right) \\ c_i &= f_i \odot c_{i-1} + u_i \odot \tilde{c}_i \\ e_i &= o_i \odot \tanh(c_i) \end{aligned} \tag{3}$$

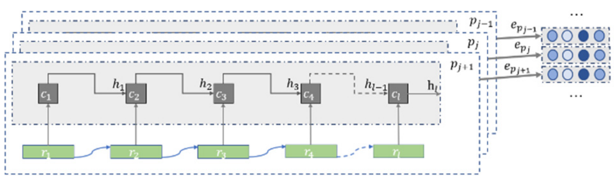


FIGURE 2. **Schematic overview of multiple-relation paths embedding.**

where $c_i \in R^{d'}$ and $\tilde{c}_i \in R^{d'}$ separately denote the memory state vector and information module, respectively; d' is the number of hidden units; and u_i, o_i, and f_i represent the input, output, and forget gate, respectively. W_c, W_f, W_u and $W_o \in R^{d' \times 3d}$ are mapping coefficient matrices, while b_c, b_f, b_u, b_o are bias vectors. $\odot$ is the elementwise product of two vectors, and $\sigma(\cdot)$ is the activation function set as a sigmoid. After processing all terms of the relation path p, we obtained the last state e_l to represent entity e_{p_j}, which is the target entity of the l-length relation path p_j.

Having established the representation of paths e_p, we aim to predict whether there is a relation path p between an entity pair (e_s, e_t). Hence, for a given multiple-step relation path triple (s, p, t), its plausibility score is formulated as follows:

$$E\left(s, p, t\right) = \left\| e_p - e_r \right\| \tag{4}$$

Equation (4) calculates the probability that the path is true, and a lower score represents a higher probability. We minimized $E\left(s, r, t\right) = \|e_s + e_r - e_t\|$ with the direct relation triple (s, r, t) so that $e_r \approx e_t - e_s$.

We encourage the discrimination of valid triplets and broken triplets through the following loss function:

$$L_{KG} = \sum\nolimits_{(s,r,t)\in T}\left[L\left(s, r, t\right) + \frac{1}{Z}\sum\nolimits_{p\in P} R\left(p|s, t\right) L\left(p, r\right)\right] \tag{5}$$

where $L\left(s, r, t\right)$ and $L\left(p, r\right)$ are margin-based loss functions:

$$L\left(p, r\right) = \sum\nolimits_{(s,r',t)\in T^-}\left[\gamma + E\left(p, r\right) - E\left(p, r'\right)\right]_+ \tag{6}$$

and

$$L\left(s, r, t\right) = \sum\nolimits_{(s',r',t')\in T^-}\left[\gamma + E\left(s, r, t\right) - E\left(s', r', t'\right)\right]_+ \tag{7}$$

In (5), (6) and (7), $R\left(p|s, t\right)$ denotes the weight of the relation path p given the entity pair (s, t), as mentioned above; $Z = \sum_{p\in P} R\left(p|s, t\right)$ is a normalized factor; r is the direct relation of the entity pair (s, t); $[x]_+ = max\left(0, x\right)$ is a standard hinge loss, γ is the margin, T is the valid triples set and T^- is the invalid triplets set: $T^- = \left\{\left(s', r, t\right)\right\} \cup \left\{\left(s, r', t\right)\right\} \cup \left\{\left(s, r, t'\right)\right\}$. After performing embedding layer models, we obtain user and item representations, namely, e_u and e_i, by encoding the path semantics. The prediction layer takes these new representations as inputs to calculate the probabilities of a user clicking a candidate item.

C. PERSONALIZED RECOMMENDATION

In the last section, we introduce how to obtain user and item embeddings that encode valuable path semantic information for the recommendation. In this section, we describe details for generating a prediction list in the top-K recommendation. The input of the prediction layer in MRP2Rec is two feature vectors e_u and e_i calculated by the embedding layer that separately represents user u and item v. We compute the predicted probabilities of whether a user will click a candidate item via the following function:

$$\hat{y}_{ui} = e_u^{\mathrm{T}} e_i. \tag{8}$$

In this function, we make use of the inner product to effectively predict users' ratings via the network. After performing the prediction layer, we obtain the top-K items of the prediction results according to the score.

To optimize the prediction model, we assigned observed interaction data higher prediction values than unobserved interactions by following the loss function:

$$L_{RS} = \sum\nolimits_{u\in U, i\in I} \Gamma\left(\hat{y}_{ui}, y_{ui}\right). \tag{9}$$

where U and I represent the user set and item set, respectively, and Γ is the cross-entropy function.

D. OPTIMIZATION

The complete loss function of the MRP2Rec joint (5) and (9) is as follows:

$$L = L_{KG} + L_{RS} + \lambda \left\|W\right\|_2^2. \tag{10}$$

In this equation, the first term measures loss in the embedding layer, which was introduced previously. The loss in the prediction layer is calculated by the second term. The last item is the L_2 regularization term that aims to prevent overfitting.

We optimize L_{KG} and L_{RS} alternatively, and the mini-batch Adam [34] algorithm is used to optimize the loss of

TABLE 2. Basic statistics for the two datasets.

		MovieLens-1M	Book-Crossing
User-Item Interaction	#Users	6,036	17,860
	#Items	2,347	14,910
	#Interactions	753,772	139,746
Knowledge Graph	#Entities	6,729	24,039
	#Relations	7	10
	#Triplets	20,195	19,793

the embedding layer and the prediction layer. Adam is a widely used optimizer that can adaptively control the learning rate with respect to the absolute value of the gradient. For a batch of randomly sampled training triplets, we update all entities' embedding; then, we take a batch of (u, i) randomly and obtain the gradients of the prediction loss for model parameter updating.

IV. EXPERIMENTS

We estimate the performance of MRP2Rec in two recommendation scenarios: movie and book. In this section, we introduce the datasets, experimental setup, baselines and experimental results in turn.

A. DATASETS

The datasets used to evaluate the performance of MRP2Rec should contain side information as well as observed user-item interactions. We process two datasets that only contain user-item interactions by constructing the KG following [23]. The first dataset, MovieLens-1M,[1] is a benchmark dataset for movie recommendation systems that contains almost one million explicit ratings. The second dataset is Book-Crossing,[2] which consists of 1,149,780 explicit ratings of books. We transform these datasets into implicit feedback data in which the positively rated user items are marked with 1, and we sample an unobserved set marked with 0 for each user. The basic statistics of the two datasets are presented in Table 2.

B. EXPERIMENTAL SETUP

We implement our MRP2Rec model in TensorFlow. The dimension of entity embedding d is fixed at 64, and the batch size is fixed at 1024. We set the weight of L_2 regularization $\lambda = 10^{-6}$, the learning rate is 0.005, the margin $\gamma = 1$, and the number of hidden units d' is 64. The settings of hyperparameters are determined according to the grid search method and the optimal configuration according to the results on the validation set. In particular, a lower batch size shows higher performance while taking more time. The dimensions of the embeddings and hidden units slightly affect the experimental results. The ratio of the training set, test set, and development set on each dataset is 6:2:2. We report the average performance after 3 repeats for each experiment.

To study how the length of paths impacts the recommendation performance, we set the different bound-lengths of paths π, i.e., L = {3,5,7}, into the proposed MRP2Rec model. The length of paths is defined as the length of the longest relation path between entity pairs plus two. We find that paths with suitable lengths of five can help the model achieve better performance, as shown in Fig. 3. The probable reason is that the shorter path has a weaker ability to express features and the longer path carries more unclearly semantic meanings.

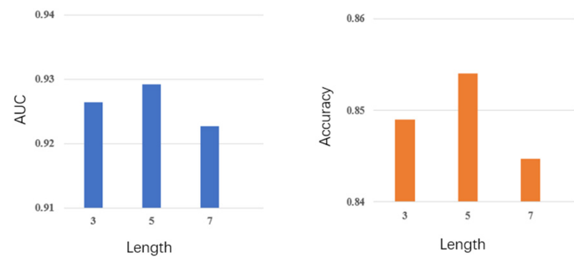


FIGURE 3. The effect of path length for movie.

We evaluate our model with the following steps. We first use the trained model to each piece of user-item interactions in each test set and obtain the output of the predicted click probability in *click-through rate* (CTR) prediction. The *AUC* and *accuracy* are calculated to evaluate the CTR prediction performance. Then, we utilize the trained model to choose K items with the highest predicted click probability for each user in the top-K recommendation. *precision@K* and *recall@K* are calculated to estimate the predicted sets.

C. BASELINES

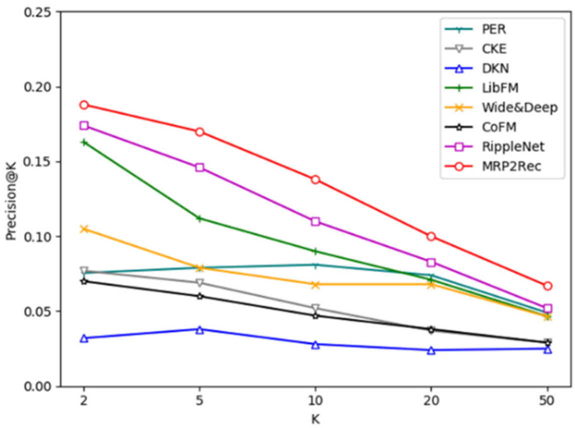
We compare our MRP2Rec model with seven state-of-the-art baselines, and the parameter setup of baselines refers to their original papers or their proposed source code.

- **Wide&Deep** [4] is a deep recommendation model combining a (deep) nonlinear channel with a (wide) linear channel. We use the representation learning method TransR [35] to learn the entity embeddings and concatenate them with the raw features of users and items. The results of the processing are the Wide&Deep input.
- **LibFM** [3] is a feature-based factorization model. The process for dealing with the input for LibFM is the same as in Wide&Deep.
- **PER** [5] treats the KG as heterogeneous information networks, and the path features between users and items are represented by manually designed meta-path-based features. We use all the user-item-attribute-item features for PER (e.g., "user-movie-director-movie") in this experiment.
- **CKE** [15] extends the CF module by combining structural, visual, and textual knowledge to represent items

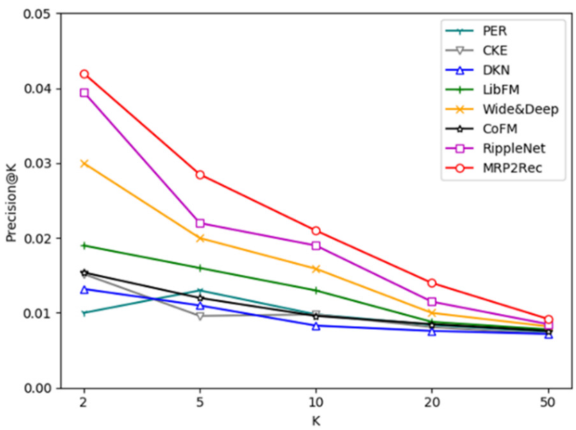
[1] https://grouplens.org/datasets/movielens/1 m/
[2] http://www2.informatik.uni- freiburg.de/~cziegler/BX/

TABLE 3. The results of AUC and *accuracy* in CTR prediction.

Model	MovieLens-1M		Book-Crossing	
	AUC	ACC	AUC	ACC
Wide&Deep	0.899	0.823	0.715	0.626
LibFM	0.893	0.810	0.687	0.638
PER	0.710	0.662	0.632	0.598
CKE	0.803	0.746	0.670	0.643
DKN	0.654	0.592	0.625	0.602
CoFM	0.810	0.726	0.709	0.658
RippleNet	**0.920**	**0.840**	**0.732**	**0.665**
MRP2Rec	**0.929***	**0.854***	**0.767***	**0.685***
%AveImprov	11.63%	11.13%	8.56%	5.21%



(a) MovieLens-1 M



(b) Book-Crossing

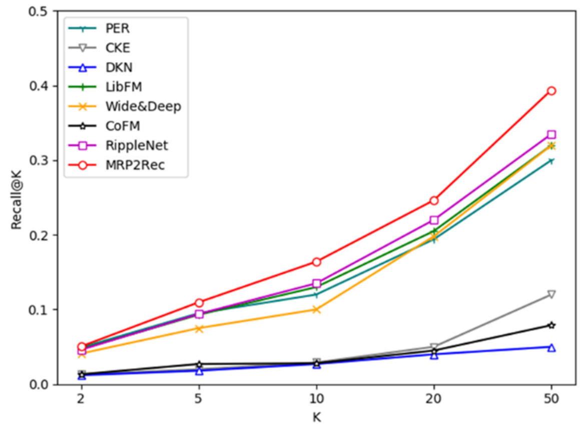
FIGURE 4. The results of *precision@K* in the top-*K* recommendation.

for recommendations. In this experiment, we implement CKE with a structural knowledge module since the datasets have no textual and visual knowledge.

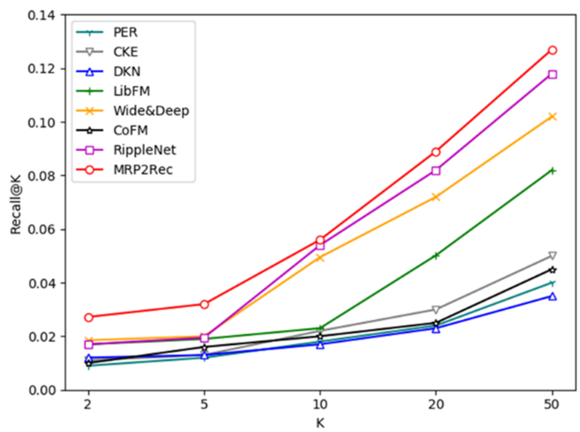
- **DKN** [17] combines multiple channels of entity embedding and text embedding into a novelly designed CNN module. Textual input for DKN is set as item names in this experiment.
- **CoFM** [20] jointly trained FM and TransE by sharing parameters of aligned entities and items.
- **RippleNet** [28] is a memory-network-like approach that learns user preferences for recommendations through preference propagation on the knowledge graph.

D. RESULTS AND ANALYSIS

The experimental results of all models in CTR prediction and top-K recommendation are reported in Table 3, Fig. 4, and Fig. 5. Several observations are described as follows:



(a) MovieLens-1 M



(b) Book-Crossing

FIGURE 5. The results of *recall@K* in the top-*K* recommendation.

Wide&Deep and LibFM show satisfactory performance, illustrating that these models can use the side information from the knowledge graph in their algorithms for an effective recommendation. However, LibFM achieves better performance in movie recommendation than book recommendation, which may be because LibFM models user-item interactions in a linear fashion, and the performance improvement is hindered by its expressive ability for exploring complex patterns in theory. PER shows better performance because path semantics can be utilized to predict user preference. It performs better in movie recommendation than book because MovieLens-1M is denser than book dataset and the manually defined meta-paths are more likely to be optimal in denser datasets. CKE does not perform as efficiently as other baselines. The reason may be that it cannot extract relation semantics from KGs, and structural knowledge in these datasets is sparse. CoFM performs poorly because these two datasets are relatively sparse, and it does not consider multiple-step relation semantics between entity pairs. DKN performs the worst since item names are too ambiguous and insufficient to provide valuable information for word embedding. RippleNet performs best among all baselines, demonstrating that RippleNet can precisely capture user interests from relation semantics for high-order relation paths in KGs.

In conclusion, our MRP2Rec model performs best among all models on the two datasets. Notably, MRP2Rec achieves an average improvement of 11.13% and 5.21% in *accuracy* in movie and book recommendations, respectively. The experimental results demonstrate that our method can exploit multiple-relation paths without manually designed features. In this way, our model can sufficiently use rich information in KGs for recommendations. The data sparsity problem is relieved by coding various relation semantics in user and item representation, and finally, the recommendation performance can be improved and exceed many state-of-the-art baselines.

V. CONCLUSION AND FUTURE WORK

Various relation semantics of multiple-step paths in knowledge graphs have received increasing attention from the recommendation community due to its effectiveness in improving recommendation performance. In this paper, we propose a novel knowledge graph-based recommendation model, MRP2Rec, with multiple-step relation path semantic embeddings for exploring different relation semantics to represent users and items. That is, MRP2Rec represents users and items not only by capturing the latent features of observed user-item interactions but also automatically by learning various semantic relations between entities encoded in KGs. Experiments on two real-world datasets verify that MRP2Rec has the ability to gain a consistent improvement in performance compared with the state-of-the-art recommendation methods. For future work, we will consider the importance of the entity types in multiple-step paths of KGs to improve our model.

REFERENCES

[1] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–45, Jul. 2014, doi: 10.1145/2556270.

[2] C. Shi, J. Liu, F. Zhuang, P. S. Yu, and B. Wu, "Integrating heterogeneous information via flexible regularization framework for recommendation," *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 835–859, Dec. 2016, doi: 10.1007/s10115-016-0925-0.

[3] S. Rendle, "Factorization machines with libfm," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 57:1–57:22, May 2012, doi: 10.1145/2168752.2168771.

[4] H.-T. Cheng, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, H. Shah, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, and W. Chai, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst. DLRS*, 2016, pp. 7–10, doi: 10.1145/2988450.2988454.

[5] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *Proc. 7th ACM Int. Conf. Web Search Data Mining WSDM*, 2014, pp. 283–292, doi: 10.1145/2556195.2556259.

[6] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, "Semantic path based personalized recommendation on weighted heterogeneous information networks," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. - CIKM*, 2015, pp. 453–462, doi: 10.1145/2806416.2806528.

[7] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu, "Recurrent knowledge graph embedding for effective recommendation," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 297–305, doi: 10.1145/3240323.3240361.

[8] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 950–958, doi: 10.1145/3292500.3330989.

[9] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, pp. 5329–5336, doi: 10.1609/aaai.v33i01.33015329.

[10] R. Catherine and W. Cohen, "Personalized recommendations using knowledge graphs: A probabilistic logic programming approach," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 325–332, doi: 10.1145/2959100.2959131.

[11] C. Luo, W. Pang, Z. Wang, and C. Lin, "Hete-CF: Social-based collaborative filtering recommendation using heterogeneous relations," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 917–922, doi: 10.1109/ICDM.2014.64.

[12] D. Shi, T. Wang, H. Xing, and H. Xu, "A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105618, doi: 10.1016/j.knosys.2020.105618.

[13] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for Top- n recommendation with a neural co-attention model," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1531–1540, doi: 10.1145/3219819.3219965.

[14] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 635–644, doi: 10.1145/3097983.3098063.

[15] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 353–362, doi: 10.1145/2939672.2939673.

[16] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009, doi: 10.1109/MC.2009.263.

[17] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," in *Proc. World Wide Web Conf. World Wide Web WWW*, 2018, pp. 1835–1844.

[18] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, "SHINE: Signed heterogeneous information network embedding for sentiment link prediction," in *Proc. 11th ACM Int. Conf. Web Search Data Mining WSDM*, 2018, pp. 592–600, doi: 10.1145/3159652.3159666.

[19] Y. Zhang, Q. Ai, X. Chen, and P. Wang, "Learning over knowledge-base embeddings for recommendation," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2018, pp. 1281–1284.

[20] G. Piao and J. G. Breslin, "Transfer learning for item recommendations and knowledge graph completion in item related domains via a co-factorization model," in *Proc. Eur. Semantic Web Conf.*, 2018, pp. 496–511, doi: 10.1007/978-3-319-93417-4_32.

[21] M. He, B. Wang, and X. Du, "HI2Rec: Exploring knowledge in heterogeneous information for movie recommendation," *IEEE Access*, vol. 7, pp. 30276–30284, 2019, doi: 10.1109/ACCESS.2019.2902398.

[22] L. Yu, G. Xu, Y. Wang, Y. Zhang, and F. Li, "ADPE: Adaptive dynamic projected embedding on heterogeneous information networks," *IEEE Access*, vol. 8, pp. 38970–38984, 2020, doi: 10.1109/ACCESS.2020.2975895.

[23] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, "Multi-task feature learning for knowledge graph enhanced recommendation," in *Proc. World Wide Web Conf. WWW*, 2019, pp. 2000–2010, doi: 10.1145/3308558.3313411.

[24] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in *Proc. World Wide Web Conf. WWW*, 2019, pp. 151–161, doi: 10.1145/3308558.3313705.

[25] X. Xin, X. He, Y. Zhang, Y. Zhang, and J. Jose, "Relational collaborative filtering: Modeling multiple item relations for recommendation," in *Proc.42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, vol. 2019, pp. 125–134, doi: 10.1145/3331184.3331188.

[26] J. Zhao, Z. Zhou, Z. Guan, W. Zhao, W. Ning, G. Qiu, and X. He, "IntentGC: A scalable graph convolution framework fusing heterogeneous information for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2347–2357, doi: 10.1145/3292500.3330686.

[27] X. Sha, Z. Sun, and J. Zhang, "Attentive knowledge graph embedding for personalized recommendation," 2019, *arXiv:1910.08288*. [Online]. Available: http://arxiv.org/abs/1910.08288

[28] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 417–426, doi: 10.1145/3269206.3271739.

[29] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proc. World Wide Web Conf. - WWW*, 2019, pp. 3307–3313, doi: 10.1145/3308558.3313417.

[30] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang, "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 968–977, doi: 10.1145/3292500.3330836.

[31] Y. Qu, T. Bai, W. Zhang, J. Nie, and J. Tang, "An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation," in *Proc. 1st Int. Workshop Deep Learn. Pract. High-Dimensional Sparse Data*, Aug. 2019, pp. 1–9, doi: 10.1145/3326937.3341257.

[32] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 705–714, doi: 10.18653/v1/D15-1082.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. ICLR Conf. Track*, Dec. 2014, pp. 1–15.

[35] H. Lin, Y. Liu, W. Wang, Y. Yue, and Z. Lin, "Learning entity and relation embeddings for knowledge resolution," *Procedia Comput. Sci.*, vol. 108, pp. 345–354, Jan. 2017, doi: 10.1016/j.procs.2017.05.045.

TING WANG is currently pursuing the master's degree with the College of Computer Science and Technology, Jilin University, China. Her research interests include knowledge graph and recommender systems.

DAQIAN SHI is currently pursuing the Ph.D. degree with the University of Trento, Italy. His research interests include neural language processing and knowledge representation.

ZHAODAN WANG received the master's degree from Jilin University. She is currently working with the Aviation University of Air Force. Her research interests include natural language processing and information extraction.

SHUAI XU (Member, IEEE) is currently pursuing the Ph.D. degree with the State Key Laboratory of Integrated Optoelectronics, College of Electronic Science and Engineering, Jilin University, China. His research interests include fabrication of micro-optics and machine learning.

HAO XU received the Ph.D. degree from the Department of Information Engineering and Computer Science, University of Trento, Italy, in 2012. He is currently a Professor with the College of Computer Science and Technology, Jilin University, China. His research interests include human-centered AI, knowledge graph, and human–computer interaction.