Knowledge Enhanced Graph Neural Networks for Explainable Recommendation

Ziyu Lyu, Yue Wu, Junjie Lai, Min Yang, Chengming Li, and Wei Zhou

Abstract—Recently, explainable recommendation has attracted increasing attentions, which can make the recommender system more transparent and improve user satisfactions by recommending products with useful explanations. However, existing methods trend to trade-off between the recommendation accuracy and the interpretability of recommendation results. In this manuscript, we propose Knowledge Enhanced Graph Neural Networks (KEGNN) for explainable recommendation. Semantic knowledge from the external knowledge base is leveraged into representation learning of three sides, respectively user, items and user-item interactions, and the knowledge enhanced semantic embedding are exploited to initialize the user/item entities and user-item relations of one constructed user behavior graph. We design a graph neural networks based user behavior learning and reasoning model to perform both semantic and relational knowledge propagation and reasoning over the user behavior graph for comprehensive understanding of user behaviors. On the top of comprehensive representations of users/items and user-item interactions, hierarchical neural collaborative filtering layers are developed for precise rating prediction, and one generation-mode and copy-mode combined generator is devised for human-like semantic explanation generation by integrating the copy mechanism into gated recurrent neural networks. Quantitative and qualitative results demonstrate the superiority of KEGNN over the state-of-art methods, and the explainability and interpretability of our method.

Index Terms—Recommender systems, explainable recommendation, graph neural networks, knowledge reasoning

1 Introduction

With the explosive growth of online information, recommender systems have played a vital role in addressing the information overload problem [1], [2]. Traditional recommender systems (RS) typically rely on collaborative filtering methods (CF) [3], [4]. Collaborative filtering leverages the history records of users in order to generate recommendations. CF methods can be divided into two categories: memory-based and model-based techniques. Memory-based methods include user-based CF and item-based CF [5]. User-based CF finds the most similar users to the target user, by comparing their vectors of item ratings using some similarity measure (e.g., cosine similarity). The predicted score of an unrated item by the target user is the similarity weighted average of other users' preferences on the item. Item-based CF takes a transposed view of user-based CF [6], [7]. Model-based methods make recommendations based on learning models such as latent factor models that employ matrix factorization [8], [9].

Recently, deep learning techniques have demonstrated its effectiveness when applied to information retrieval and recommender systems research. Many deep learning based recommendation methods have been proposed and achieved high recommendation performance [10], [11]. Deep learning models which comprise hundreds of layers and millions of parameters are complex black-box models, lacking interpretability and transparency of decision making [12]. Despite the recommendation accuracy, the recommendation explainability which clarifies why such items are recommended, has attracted increasing attentions [13]. Therefore, the new task called explainable recommendation has emerged, which can provide recommendation results with explanations to users or system designers [14]. Explainable recommendation not only improves the transparency, interpretability, trustworthiness of the recommender systems, but also improves user satisfaction [15], [16].

Earlier explainable recommendation methods were generally based on matrix factorization techniques and aligned latent factors with pre-defined item attributes/features. The pre-defined item attributes will be utilized to provide feature-level template explanations. For example, Zhang *et al.* proposed an explicit factor model (EFM) [17] which aligned latent factors with predefined item features for explainable recommendation. He *et al.* [18] extracted item aspects from textual reviews, and modeled the user-item-aspect ternary relation as a tripartite graph. They proposed a generic algorithm TriRank for ranking on the tripartite graph and performed personalized recommendation. The extracted aspects can explain the recommendation results. However, those feature/aspect-level explainable methods required some human efforts such

- *Ziyu Lyu and Min Yang are with the Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Sciences Beijing, 100045, China. E-mail: {zy.lv, min.yang}@siat.ac.cn.*
- *Yue Wu is with the Chongqing University of Posts and Telecommunications, Chongqing 400065, China. E-mail: yue.wu@siat.ac.cn.*
- *Junjie Lai is with the University of Science and Technology of China, Hebei 101127, China. E-mail: laijj94@mail.ustc.edu.cn.*
- *Chengming Li is with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510275, China. E-mail: lichengming@mail.sysu.edu.cn.*
- *Wei Zhou is with the Chongqing University, Chongqing 400044, China. E-mail: zhouwei@cqu.edu.cn.*

Manuscript received 5 February 2021; revised 6 December 2021; accepted 7 January 2022. Date of publication 13 January 2022; date of current version 3 April 2023.
This work was supported by the National Natural Science Foundation of China under Grants 62002352 and 72074036.
(Corresponding author: Wei Zhou.)
Recommended for acceptance by L. Zou.
Digital Object Identifier no. 10.1109/TKDE.2022.3142260

as predefined attributes/features, sentiment analysis. And the generated template-based explanations lacked flexibility.

With the development of deep learning methods and natural language processing techniques, several explainable recommendation methods have been proposed by leveraging natural language generation techniques to generate natural textual explanations when providing personalized recommendations. For example, Chen *et al.* [19] designed a novel attention mechanism to explore the usefulness of reviews and proposed a neural attentional regression model for rating prediction with review-level explanations (NARRE). Li *et al.* [20] proposed a deep learning based framework named Neural Rating Regression (NRT) which simultaneously performed rating prediction, and adopted the sequence-to-sequence (seq2seq) framework [21] based gated recurrent neural networks [22] to generate abstractive tips. However, it is difficult to generate high-quality textual explanations due to the data sparsity.

As knowledge graph can involve more facts and connections [23], [24], some researchers have leveraged knowledge graph for recommendation and enhanced the recommendation explainability via the graph reasoning paths [25], [26], [27]. For example, Wang *et al.* [25] proposed an end-to-end knowledge graph aware recommendation framework RippleNet in which knowledge graph has been incorporated to stimulate the propagation of user preferences over the set of knowledge entities and discover users' hierarchical potential interests by iteratively propagating users' preferences in the knowledge graph. The graph reasoning paths enhance the recommendation explainability. Wang *et al.* [27] proposed a novel model named Knowledge-aware Path Recurrent Network (KPRN) model, which generated path representations by composing the semantics of both entities and relations, and inferred user preferences through effective reasoning on paths. The attention mechanism has been utilized to offer path-wise explanations. However, the graph reasoning paths are not intuitive for users, and the knowledge graph might involve redundant entities which causes homogeneous recommendation results.

In this manuscript, we propose novel **K**nowledge **E**nhanced **G**raph **N**eural **N**etworks (KEGNN) for explainable recommendation. KEGNN leverages the semantic knowledge from the external knowledge base to learn knowledge enhanced semantic embedding of three sides: respectively users, items and user-item interactions. From the user-item interactions, we construct the user behavior graph and initialize the user behavior graph with the knowledge enhanced semantic embedding. A graph neural networks based user behavior learning and reasoning model is proposed to comprehensively understand user behaviors, by propagating user preferences through connected users/items and perform multi-hop reasoning over the user behavior graph. Finally hierarchical collaborative filtering layers are developed for rating prediction, and the copy mechanism is incorporated with gated recurrent units (GRU) based generator to generate human-like semantic explanations. The main contributions are summarized as follows:

- We integrate the semantic concepts from the external knowledge into the hierarchical semantic representation learning for users, items and user-item interactions, and exploit the knowledge enhanced semantic representation to initialize user/item nodes and user-item relations of the user behavior graph constructed from user-item interactions.
- A graph neural networks based user behavior learning and reasoning model is devised to perform knowledge reasoning over the user behavior graph. Both *semantic preferences* and *relational preferences* among user behaviors are explored for comprehensive understanding of user preferences.
- We design hierarchical neural collaborative filtering layers for precise rating prediction. Especially, one *relation-aware neural layer* is developed to incorporate the user-item relational preferences into user-item interaction prediction.
- The explanation generation module combines the *generation mode* and the *copy mode* for high-quality human-like semantic explanations, by integrating he copy mechanism into the sequence decoding process.
- Extensive experiments have been conducted on three real datasets. The experiment results show our proposed method has superior recommendation performance than the-state-of-art methods. Both quantitative and qualitative experiments demonstrate the high-quality explainability and interpretability of our method.

The rest of this manuscript is organized as follows: We formulate our problem in Section 2. Section 3 illustrates details of our proposed method. Experimental setup and experimental results are respectively demonstrated in Sections 4 and 5. Related work is presented in Section 6. Section 7 concludes this manuscript and provides directions for future work.

2 Problem Formulation

We formally state the problem definition. Table 1 summarizes the notations and key concepts used in this manuscript. We use lower-case fonts for scalars, bold lower-case fonts for vectors and bold upper-case fonts for matrices. For example, p is a scalar, $\mathbf{p}$ is a vector and $\mathbf{P}$ is a matrix.

We have a set of user U and a set of items V. The number of users is m and the number of items is n. The user-item rating matrix is $\mathbf{R} \in \mathbb{R}^{m\times n}$. Each non-zero entry r_{ij} indicates the observation of the interaction between the user u_i and v_j (e.g., rating score). In addition, each rating r_{ij} is associated with the textual review $d_{ij} = [w_1, w_2, \ldots, w_{|d_{ij}|}]$, which demonstrates the feelings or comments on the item v_j from the user u_i. w indicates the context word in reviews, and $|d_{ij}|$ is the length of the review document d_{ij}. The review collection is denoted as D, and the vocabulary of the review collection is V_d. And the commonsense knowledge base $\mathcal{KB}$ (e.g., ConceptNet [28]) is utilized to enhance semantic representation learning of users, items, and user-item interactions.

Definition 1 (Problem Definition). *Given the users U, items V, user-item interaction matrix $\mathbf{R}$, the review collection D, and the knowledge graph $\mathcal{KB}$, the problem is to predict the rating $\hat{r}_{ij}$ from the given user u_i for the recommended item v_j (unrated items) and generate the textual explanations $Y_{ij} = [w_{ij1}, w_{ij2}, \ldots, w_{ij|Y|}]$.*

TABLE 1
Notations

Symbols	Description
U, V	the set of users and items
$\mathbf{R}$	user-item rating matrix
$r_{ij}, \hat{r}_{ij}$	the real rating/predicted rating of user u_i to item v_j
d_{ij}	the textual review associated with r_{ij}
D	the collection of all reviews
d^u , d^v	the aggregated documents for user or item
w, $\mathbf{w}$	the word token, and the word embedding
V_d	vocabulary of the review collection
$\mathcal{KB}$	common-sense knowledge base
Y, y	the generated explanation, explanation word
$\mathbf{C}^g, \mathbf{C}^s$	global-level/sentence-level contextualized representation
$\mathbf{C}^h$	hierarchical contextualized representation
c, $\mathbf{c}$	retrieved concept from KB, concept embedding
$\mathbf{K}$	knowledge-aware representation
$\mathbf{x}^u, \mathbf{x}^v$	knowledge-enhanced representation for user, item
$\mathbf{x}^{uv}$	knowledge-enhanced representation for user-item interaction
$\mathcal{G}$	the constructed user behavior graph in Section 3.2
$\mathbf{e}^u$, $\mathbf{e}^v$	user/item node representation in $\mathcal{G}$
$\mathbf{r}$	relation representation in $\mathcal{G}$
$\mathcal{N}_{e_i}$, $\mathbf{E}$	ego-network of node e_i, ego-network representation
ϕ	aggregation function in graph neural networks
$\mathbf{W}$, b	weight matrix, bias term

3 Method

We propose knowledge enhanced graph neural networks for explainable recommendation. The architecture of the proposed method is shown in Fig. 1. It mainly consists of four modules: respectively knowledge enhanced semantic representation learning, graph neural networks based user behavior learning and reasoning, hierarchical collaborative filtering, and textual explanation generation. The knowledge enhanced semantic representation learning module is designed to learn the hierarchical semantic representation learning of users, items, and user-item interactions, enhanced by the semantic concepts from the knowledge base (Section 3.1). The user behavior learning and reasoning module is for modelling user behaviors and learning the underlying relations among the constructed user behavior graph, and obtains comprehensive representations of users/items and the relational representation of user-item interactions through multi-hop knowledge reasoning (Section 3.2). On the top of the user behavior learning and reasoning, we design one hierarchical collaborative filtering module for precise rating prediction (Section 3.3) and one explanation generation module for explanation generation (Section 3.4). The explanation generation module combines both the gated recurrent unit (GRU) based generator and the copy mechanism to generate high-quality human readable explanations. We will illustrate details of each module in the following sections.

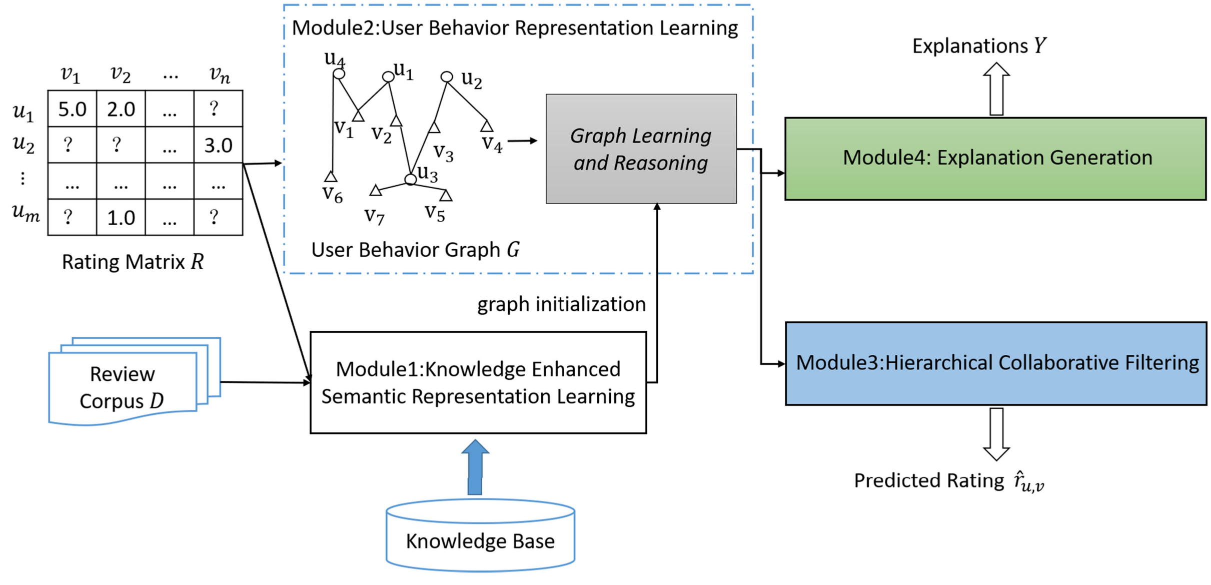
3.1 Knowledge Enhanced Semantic Representation learning

In order to learn the semantic representation learning of users, items and user-item interactions, we perform time-order document pooling for users and items. For a user u or a item v, we aggregate history reviews associated with the item or commented by the user in time-order as the user/item review documents d^u/d^v, according to the review time associated with each review in dataset. The aggregated documents might have semantic sequential dependencies through time-order reviews. Therefore, we have three types of textual documents for users d^u, items d^v and user-item interactions d^{uv}. Further knowledge enhanced semantic representation learning is performed on the three types of documents, and Fig. 2 shows the architecture of knowledge enhanced semantic representation learning module.

3.1.1 *Context Representation*

Word-level Embedding. Each document d (any type of document), we first have the word embedding sequence as $[\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{|d|}]$. $\mathbf{w}_i \in \mathbb{R}^{d_w}$ is the word embedding for the word w_i generated from pre-trained embedding such as word2-vec embedding [29], and d_w denotes the size of the word embedding.

Contextualized Representation. On the top of the word embedding sequence $[\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{|d|}]$ of each document d,



	v_1	v_2	...	v_n
u_1	5.0	2.0	...	?
u_2	?	?	...	3.0
⋮	...	...	...	...
u_m	?	1.0	...	?

Rating Matrix R
Module2:User Behavior Representation Learning
u4
u1
u2
v1
v2
v3
v4
u3
v6
v7
v5
User Behavior Graph G
Graph Learning and Reasoning
Explanations Y
Module4: Explanation Generation
graph initialization
Review Corpus D
Module1:Knowledge Enhanced Semantic Representation Learning
Module3:Hierarchical Collaborative Filtering
Predicted Rating $\hat{r}_{u,v}$
Knowledge Base

Fig. 1. The architecture of framework.

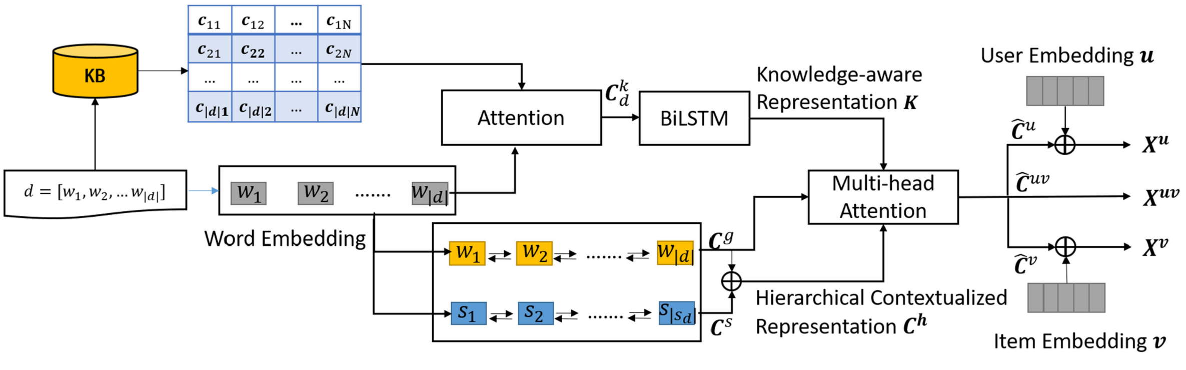

KB
c_{11} c_{12} ... c_{1N}
c_{21} c_{22} ... c_{2N}
... ...
$c_{|d|1}$ $c_{|d|2}$ ... $c_{|d|N}$
$d = [w_1, w_2, ... w_{|d|}]$
w_1 w_2 .... $w_{|d|}$
Word Embedding
Attention
C_d^k
BiLSTM
Knowledge-aware Representation K
w_1 w_2 .... $w_{|d|}$
s_1 s_2 .... $s_{|s_d|}$
C^g
C^s
Hierarchical Contextualized Representation C^h
Multi-head Attention
User Embedding u
$\hat{C}^u$
X^u
$\hat{C}^{uv}$
X^{uv}
$\hat{C}^v$
X^v
Item Embedding v

Fig. 2. Knowledge enhanced semantic representation learning.

one bidirectional Long Short-Term Memory network (BiLSTM) [30] is utilized to capture the global contextualized embedding $\mathbf{C}_d^g = BiLSTM([\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{|d|}])$. In addition, we consider the sentence-level contextualized embedding. For each sentence, we sum up the word embedding of the sentence $\mathbf{s}_i = \sum_j^{|s_i|} \mathbf{w}_j$, and obtain the sentence representation $\mathbf{s}_i$ for the ith sentence of the document. Then, we have the sequence of sentence representation $\mathbf{S}_d = [\mathbf{s}_1, \mathbf{s}_1, \ldots, \mathbf{s}_{s_d}]$. s_d is the number of sentences in the document d. Another BiLSTM is adopted on the sentence representation sequence to have the sentence-level contextualized embedding $\mathbf{C}_d^s = BiLSTM(\mathbf{S}_d)$. We concatenate the global-level contextualized embedding and the sentence-level contextualized embedding to have the *hierarchical contextualized embedding* $\mathbf{C}_d^h = [\mathbf{C}_d^g, \mathbf{C}_d^s]$.

3.1.2 Knowledge-Aware Representation

We leverage the knowledge base to enhance the semantic representation learning. For each word w_{di} of the review document d, we retrieve top-N relevant concepts from the knowledge base. $\mathbf{c}$ denotes the embedding of the knowledge concept. The attention mechanism [31] is employed to learn different relevance of the top-N concepts. The knowledge-aware embedding $\mathbf{c}_{di}^k$ for the given word w_{di} is obtained as follows:

$$\mathbf{c}_{di}^k = \sum_j^{|N|} \alpha_{dij} \mathbf{c}_{dij}^k$$
$$\alpha_{dij} = softmax(\mathbf{O}_{dij})$$
$$\mathbf{O}_{dij} = \tanh(\mathbf{W}_w \mathbf{w}_{di} + \mathbf{W}_c \mathbf{c}_{dij} + \mathbf{b}^k) \quad (1)$$

As the top-1 concept is the raw word, we directly feed the knowledge-aware embedding $\mathbf{c}^k$ sequence into one BiLSTM layer $\mathbf{C}_d^k = [\mathbf{c}_1^k, \mathbf{c}_2^k, \ldots, \mathbf{c}_{|d|}^k]$ (without further concatenation operation), and obtain the *knowledge-aware representation* $\mathbf{K}_d = BiLSTM(\mathbf{C}_d^k)$.

3.1.3 Knowledge Enhanced Semantic Representation

The multi-head attention [31] is adopted to further fuse the knowledge-aware representation and the contextualized representation. Multi-head attention can learn the knowledge representation over different semantic spaces, and obtain the deep knowledge enhanced semantic representation. We treat the fused representation $\mathbf{C}_d^f = sum(\mathbf{C}_d^g, \mathbf{K}_d)$ as keys, the knowledge-aware representation $\mathbf{K}_d$ as values, and the hierarchical contextualized representation $\mathbf{C}_d^h$ as query. Through the multi-head attention, we obtain the advanced semantic representation $\hat{\mathbf{C}}_\mathbf{d}$ as in Equation

$$\hat{\mathbf{C}}_\mathbf{d} = \mathbf{W}^h [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_h]$$
$$\mathbf{h}_i = Attention(\mathbf{C}_d^h \mathbf{W}_i^Q, \mathbf{C}_d^f \mathbf{W}_i^K, \mathbf{K}_d \mathbf{W}_i^V) \quad (2)$$

where $\mathbf{W}_i^h$, $\mathbf{W}_i^Q$, $\mathbf{W}_i^K, \mathbf{W}_i^V$ are the learning parameters of multi-head attention. h is the number of head. Therefore, for the three types of textual documents d^u, d^v, d^{uv}, we perform the same semantic learning process, and have the advanced semantic representation $\hat{\mathbf{C}}^u$, $\hat{\mathbf{C}}^v$, $\hat{\mathbf{C}}^{uv}$.

In addition, we represent the users/items with one-hot encoding, and a fully-connected layer is adopted to map the sparse one-hot representation into a dense representation as the user/item inherent representation $\mathbf{u}/\mathbf{v}$. For a user u_i, we combine its advanced semantic representation $\hat{\mathbf{C}}_i^u$, and the user inherent representation $\mathbf{u}_i$ to have the *knowledge enhanced semantic representation* $\mathbf{x}_i^u$[1]. Similarly, we combine the item advanced semantic representation $\hat{\mathbf{C}}_j^v$, and the item inherent representation $\mathbf{v}_j$ to have the *knowledge enhanced semantic representation* $\mathbf{x}_i^v$ for item v_j. The *knowledge enhanced semantic representation* of user-item interactions $\mathbf{x}_{ij}^{uv}$ (between user u_i and item v_j) is set as $\hat{\mathbf{C}}_{ij}^{uv}$. The knowledge-enhanced representations will be used as the initial embedding of user/item node and user-item relation in the user behavior graph in Section 3.2.

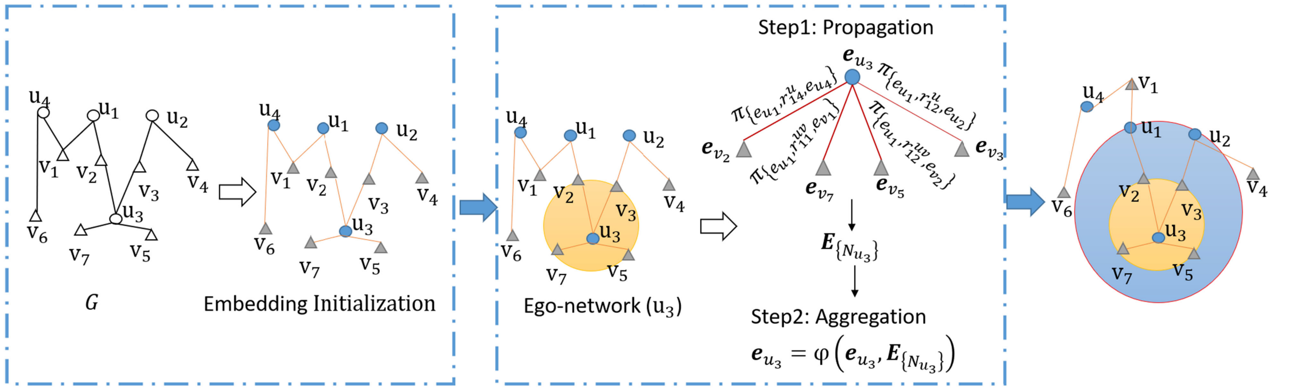
3.2 User Behavior Learning and Reasoning

For the purpose of comprehensively understanding user preferences, we devise a graph neural networks based user behavior learning and reasoning module. Fig. 3 illustrates this detailed process.

3.2.1 User Behavior Graph Construction

From the user-item interaction relationships, we construct the user behavior graph $\mathcal{G}$, in which nodes include the users and the items, and the edges denote the user-item interactions, namely $\mathcal{G} = \{(h, r, t) | h \in U, t \in V, r$ exists when the rating $r_{ht} \neq 0\}$. The node set is denoted as $\xi = \{U \cup V\}$, and the relation set is denoted as Υ.

[1] The combination function can be linear combination, e.g., sum aggregation or nonlinear combination e.g., using multilayer perceptron layer on the concatenation of the embedding. The reported results used sum aggregation.


u4
u1
u2
v1
v2
v3
v4
u3
v6
v7
v5
G
Embedding Initialization
Ego-network (u3)
Step1: Propagation
e_{u_3}
$\pi_{\{e_{u_1},r_{14}^{u},e_{u_4}\}}$
$\pi_{\{e_{u_1},r_{12}^{u},e_{u_2}\}}$
$\pi_{\{e_{u_1},r_{11}^{uv},e_{v_1}\}}$
$\pi_{\{e_{u_1},r_{12}^{uv},e_{v_2}\}}$
e_{v_2}
e_{v_3}
e_{v_7}
e_{v_5}
$E_{\{N_{u_3}\}}$
Step2: Aggregation
$e_{u_3} = \varphi\left(e_{u_3}, E_{\{N_{u_3}\}}\right)$

(a) User Behavior Graph (b) Information Propagation Layer (c) Multi-hop Reasoning

Fig. 3. User behavior learning and reasoning.

We exploit the knowledge enhanced semantic representation to initialize the node representation and the edge representation of the user behavior graph. The user node embedding $\mathbf{e}_i^u$ is initialized as $\mathbf{e}_{u_i} = \mathbf{x}_i^u$, and the item node embedding $\mathbf{e}_{v_j}$ is initialized as $\mathbf{e}_{v_j} = \mathbf{x}_j^v$. For the user-item interaction relation embedding, is initialized as $\mathbf{r}_{ij} = \mathbf{x}_{ij}^{uv}$.

3.2.2 Information Propagation Layer

Based on the architecture of graph convolution network [24], [32], we design GNN-style information propagation layer to capture the higher-order structural proximity of user behavior graph. First, we find the neighbours of one node and consider the first-order information propagation among the *ego-network* [33], [34] as shown in Fig. 3b (The node u_3 is used as an example to illustrate this process). The ego-network is denoted as $\mathcal{N}_{e_i} = \{(e_i, r, e_j) | (e_i, r, e_j) \in \mathcal{G}\}$, in which e_i is the head node. To characterize the first-order proximity structure of the node e_i, the combination of its ego-network is computed as follows:

$$\begin{aligned}\pi_{\{e_i,r_{ij},e_j\}} &= \mathbf{W}_t\sigma(\mathbf{W}_a\mathbf{e}_i + \mathbf{W}_b\mathbf{r}_{ij} + \mathbf{W}_c\mathbf{e}_j + b)\\ \beta_{ij} &= \frac{exp(\pi_{\{e_i,r_{ij},e_j\}})}{\sum_{(e_i,r_{ij},e_j)\in\mathcal{N}_{e_i}} exp(\pi_{\{e_i,r_{ij},e_j\}})}\\ \mathbf{E}_{\mathcal{N}_{e_i}} &= \sum_{(e_i,r_{ij},e_j)\in\mathcal{N}_{e_i}} \beta_{ij}\mathbf{e}_j\end{aligned} \tag{3}$$

$\pi_{\{e_i,r_{ij},e_j\}}$ measures the information propagating from the head node e_i to the connected node e_j conditioned on the relation r_{ij}. σ indicates the nonlinear activation function, e.g., tanh, ReLU. β is the normalization attention scores by normalizing $\pi_{\{e_i,r_{ij},e_j\}}$ over all the triplets in the ego-network of node e_i. Finally, the information representation propagated from the ego-network $\mathbf{E}_{\mathcal{N}_{e_i}}$ is obtained, by aggregating the representations of neighbours in the ego-network with β.

After obtaining the first-order information propagation, the next step is to aggregate the node representation $\mathbf{e}_i$, and its ego network representation $\mathbf{E}_{\mathcal{N}_{e_i}}$ with aggregation function $\phi(\mathbf{e}_i, \mathbf{E}_{\mathcal{N}_{e_i}})$. As sum aggregator has superior performance [24], [33], we adopt the sum aggregator to sum up the two representations and utilize a nonlinear transformation as follows:

$$\phi(\mathbf{e}_i, \mathbf{E}_{\mathcal{N}_{e_i}}) = \sigma(\mathbf{W_1}(\mathbf{e}_i + \mathbf{E}_{\mathcal{N}_{e_i}}) + \mathbf{b_1}) \tag{4}$$

where $\mathbf{W_1}, \mathbf{b_1}$ are the trainable weights and bias. σ is the nonlinear function such as ReLU.

3.2.3 Multi-hop Reasoning

In order to explore the high-order proximity, we perform multi-hop reasoning by stacking multiple propagation layers, and recursively gather information transferred from multi-hop neighbors. In the lth iteration, the node representation is defined as in Eq. (5).

$$\mathbf{e}_i^l = \phi(\mathbf{e}_i^{l-1}, \mathbf{E}_{e_i}^{l-1}) \tag{5}$$

where $\mathbf{e}_i^{l-1}$ is the representation of the node $\mathbf{e}_i$ obtained from previous propagation layers, gathering message passing from l-1 hop neighbours. Fig. 3c shows the example of 2-hop reasoning.

After performing multi-hop reasoning, we obtain the node (user/item) representation and relation representation from the final layer, and they will be further used for rating prediction and explanation generation.

3.3 Hierarchical Collaborative Filtering

We propose one hierarchical collaborative filtering module for precise rating prediction, as shown in Fig. 4a. In the hierarchical collaborative filtering module, we design three neural collaborative filtering layers to hierarchically perform user-item interaction prediction. In the first neural collaborative layer, we concatenate the user representation $\mathbf{e}_{u_i}$ and the item representation $\mathbf{e}_{v_j}$ obtained from the user behavior graph learning and reasoning (Section 3.2). The nonlinear transformation function f_1 is used for user-item interaction transformation as in Eq. (6).

$$\mathbf{I}_{ij} = f_1(\mathbf{W}_1[\mathbf{e}_{u_i}, \mathbf{e}_{v_j}] + b) \tag{6}$$

where $\mathbf{W}_1, b$ are the trainable parameters. f_1 is the activation function, and we use ReLU. $\mathbf{I}_{ij}$ represents the first level user-item interaction representation.

In the second layer, we devise one *relation-aware neural layer*, by integrating the user-item relation representation into user-item interaction prediction. First, the interaction representation $\mathbf{I}_{ij}$ and the user-item relation representation $\mathbf{r}_{u_i,v_j}$[2]

[2] We set $\mathbf{r}_{u_i,v_j} = \mathbf{e}_{u_i} \odot v_j$ for unobserved user-item interaction in the test phrase.

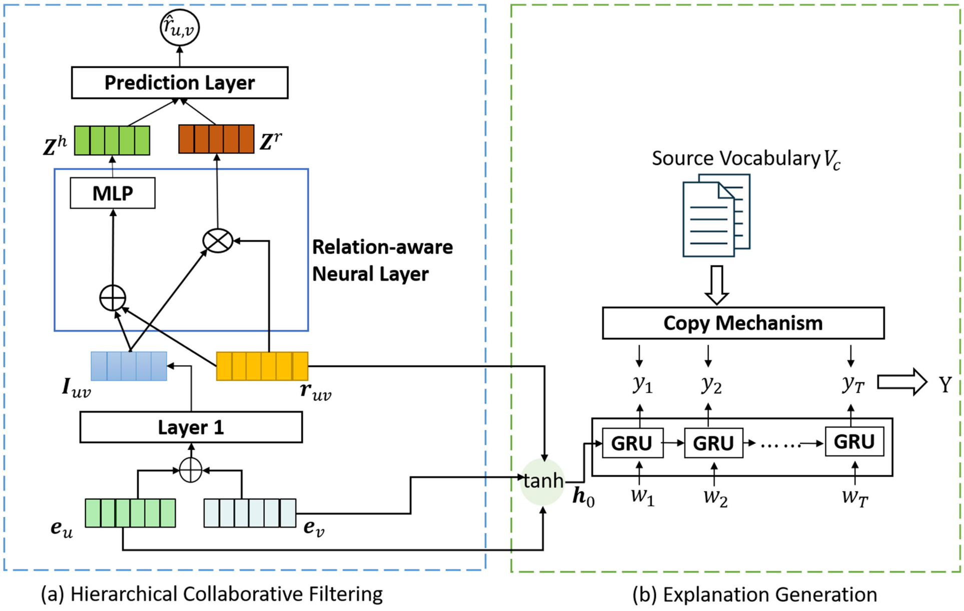


Fig. 4. Rating prediction and explanation generation.

are combined through the Hadamard product ($\odot$). Then, we obtain the *relation-aware user-item interaction representation* $\mathbf{Z}^r_{ij}$ as follows:

$$\mathbf{Z}^r_{ij} = \mathbf{I}_{ij} \odot \mathbf{r}_{u_i,v_j} \tag{7}$$

In addition, we also exploit multiple layer perception (MLP) to further fuse the user-item interaction and the relational representation, and obtain the *high-level interaction representation* $\mathbf{Z}^h_{ij}$ as follows:

$$\mathbf{Z}^h_{ij} = MLP([\mathbf{I}_{ij}, \mathbf{r}_{u_i,v_j}]) \tag{8}$$

Finally, the last neural layer takes the relation-aware user-item interaction representation and the high-level interaction representation as inputs, and predict the rating r_{ij} of the item v_j from the user u_i. The computation is defined as in Eq. (9).

$$\hat{r}_{ij} = \sigma(\mathbf{W}_l[\mathbf{Z}^r_{ij}, \mathbf{Z}^h_{ij}]) \tag{9}$$

$\mathbf{W}_l$ is the trainable parameter. $\hat{r}_{ij}$ is the predicted rating for the item v_j from the user u_i.

3.4 Explanation Generation

We develop a novel textual explanation generation module to generate high-quality human readable explanations, by combining the generation-based model and the copy mechanism (generation mode and copy mode). The right part of Fig. 4 demonstrates the details of this module. We adopt the Gate Recurrent Unit (GRU) [22] to generate textual explanation. In addition, the copy mechanism [35] is integrated with the GRU generator to select part segments from the original text source. The vocabulary in the copy mode V_c is the union set of words appearing in the user review document d_u and the item review document d_v, namely $V_c = d_u \cup d_v$.

We fuse the user representation $\mathbf{e}_u$, the item representation $\mathbf{e}_v$, and the user-item relational representation $\mathbf{r}_{uv}$ through one tanh transformation as in Eq. (10). $\mathbf{h}_0$ works as the initial state in the decoding process.

$$\mathbf{h}_0 = tanh(\mathbf{W}_u\mathbf{e}_u + \mathbf{W}_v\mathbf{e}_v + \mathbf{W}_r\mathbf{r}_{uv}) \tag{10}$$

$\mathbf{W}_u$, $\mathbf{W}_v$, and $\mathbf{W}_r$ are the trainable parameters.

3.4.1 Generator

In the generation mode, GRU is adopted as the generator. $\mathbf{h}^g_t$ is the hidden state from GRU at tth step, and $\mathbf{h}^g_t = GRU(\mathbf{h}^g_{t-1}, \mathbf{w}_t)$. When recommending the item v_j to the given user u_i, the word generation probability $P_{gen}(y_{ijt} = w_t)$ at tth step is computed as follows:

$$P_{gen}(y_{ijt} = w_t) = softmax(\mathbf{h}^g_t), w_t \in V_g \tag{11}$$

w_t is from the generation vocabulary $V_g = V_d/V_c$ which includes the remaining words not in the source vocabulary, and $\mathbf{w}_t$ is the word embedding for the word w_t.

3.4.2 Copy Mechanism

In addition, we utilize the copy mechanism to integrate the copy mode into explanation generation. The copy mode allows to select the original words from the source reviews from the user d_u or the item d_v. The word generation probability at the tth step at the copy mode $P_{copy}(y_{ijt} = w_t)$ is computed as follows:

$$P_{copy}(y_{ijt} = w_t) = softmax(\mathbf{w}_t\mathbf{W}^c[\mathbf{h}^g_t, \mathbf{h}^0_t]), w_t \in V_c \tag{12}$$

$\mathbf{W}^c$ is the learnable parameters.

Combing the generation mode and copy mode, the generation of the final explanation word y_{ijt} is defined as in Eq. (13), when predicting the rating r_{ij} on the item v_j from the given user u_i.

$$p(y_{ijt} = w_t) = p_{gen}(y_{ijt} = w_t) + p_{copy}(y_{ijt} = w_t) \tag{13}$$

The generated explanation is $Y_{ij} = [Y_{ij1}, Y_{ij2}, \ldots, Y_{ijT}]$, and T is the length of the generated explanation.

3.5 Model Optimization

The rating prediction loss $\mathcal{L}_r$ is defined as follows:

$$\mathcal{L}_r = \frac{1}{2N}\sum_{i=1}^{|U|}\sum_{v_j\in I_{u_i}^{+}}(r_{ij}-\hat{r}_{ij})^2 \tag{14}$$

where $I_{u_i}^{+}$ indicates the interacted items of the user u_i.

The explanation generation loss is defined as follows:

$$\mathcal{L}_g = \frac{1}{|I^{+}|}\sum_{(u_i,v_j)\in I^{+}}\sum_{t=1}^{T} -p(y_{ijt})logp(w_{ijt}) \tag{15}$$

I^{+} represents all the non-zero entry in the rating matrix, with which the textual review d_{ij} is associated. y_{ijt} is the generated word while w_{ijt} is the real word in the review document d_{ij}. T is the length of the generated textual explanations.

We utilize the multi-task learning [36] way to perform model optimization. One unified objective function is defined in Eq. (16), by taking the losses from both rating prediction and explanation generation. The model optimization is achieved by minimizing the unified objective function.

$$\mathcal{L} = \lambda_r\mathcal{L}_r + \lambda_g\mathcal{L}_g + \lambda||\Theta||^2 \tag{16}$$

$\mathcal{L}_r$ is the loss of rating prediction as in Eq. (14), and $\mathcal{L}_g$ is the loss of rating prediction as in Eq. (15). Θ indicates all the learnable parameters. λ_r, λ_g are the weights to control the influences of different parts. λ are the regularization weights.

4 Experimental Setup

4.1 Datasets

We use three datasets from Amazon 5-core[3] , respectively Electronics, Home & Kitchen and Music-Instruments. The rating range is in $[0,5]$. Each record in the Amazon dataset contains the "summary" filed with the rating. We regard the textual summary as the textual reviews (in the training phrase), and as the ground-truth explanations (in the test phrase). We preprocess the data by removing users whose rating records are less than 5, removing stopwords and filtering out words with low frequency. The statistics of datasets after preprocessing are listed in Table 2.

The three datasets have different scales, respectively large, medium, and small. There are 192,402 users, 63,001 items, and 1,689,188 user-item interaction records in the Electronics dataset. The vocabulary size is 70,233. In Home & Kitchen dataset, there are 66,519 users, 28,237 items and 551,682 user-item interaction records. The vocabulary size is 27,348. In Music-Instruments, there are 1,429 users, 900 items, and 10,261 user-item interaction records. The vocabulary size is 2,783. For all datasets, we randomly selected 80% user-item interactions in each data set as the training set, and 10% of user-item interactions as the test set. The remaining 10% of user-item interactions are seen as the validation set.

TABLE 2
Statistics of Datasets

Dataset	Electronics	Home & Kitchen	Musical Instruments
users	192,402	66,519	1,429
items	63,001	28,237	900
reviews	1,689,188	551,682	10,261
$\|V_d\|$	70,233	27,348	2,783

4.2 Evaluation Metrics

We evaluate both the recommendation accuracy and interpretability of generated explanations. We use the widely used metric Rooted Mean Square Error (RMSE) and Mean Absolute Error (MAE) to evaluate the performance of rating prediction. Given the predicted rating $\hat{r}_{ij}$ and the ground-truth rating r_{ij} from the user u_i for the item v_j, RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u_iv_j}(r_{ij}-\hat{r}_{ij})^2} \tag{17}$$

N is the number of ratings between users and items. A lower RMSE indicates the better performance. MAE is computed as follows:

$$MAE = \frac{1}{N}\sum_{u_iv_j}|r_{ij}-\hat{r}_{ij}| \tag{18}$$

Similar to [20], we use ROUGE [37] to evaluate the recommendation explainability. ROUGE is a classical evaluation metric for text summarization [38], which measures the number of overlapping units between the generated summarization and the ground truth text. We use ROUGE to count the overlapping units between the generated explanations and the real reviews. The definition is defined as follows:

$$ROUGE(Y_{ij}) = \frac{\sum_{g_n\in d_{ij}} Count_{match}(g_n)}{\sum_{g_n\in\hat{s}} Count(g_n)} \tag{19}$$

$Count_{match}(g_n)$ is the maximum number of n-grams co-occurring at the generated context Y and the ground-truth review document d. $Count(g_n)$ is the n-grams in $\hat{s}$. When $\hat{s}$ equals to the ground-truth review document d, we have ROUGE-Recall. When $\hat{s}$ equals to the generated context Y, we have ROUGE-Precision.

4.3 Baselines

We compare our proposed method KEGNN with the-state-of-art methods. The details of the compared methods are listed as follows:

- *Collective Topic Regression (CTR)* [39]: is a one-class collaborative filtering method, with leveraging the textual reviews to assist the collaborative filtering.
- *Probabilistic Matrix Factorization (PMF)* [9]: PMF models latent factors of users and items through Gaussian distribution and it is a classic method for rating prediction.
- *NARRE* [19]: is a neural attentional regression model for rating prediction and employs the attention mechanisms to select the useful reviews as explanations.
- *NRT* [20]: In NRT, one neural network based rating regression model is proposed for rating prediction and one gated neural network is employed to generate

[3] http://jmcauley.ucsd.edu/data/amazon

TABLE 3
Performance for Rating Prediction

Method	Electrtonics		Home & Kitchen		Music-Instruments	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
CTR	0.858	1.135	0.737	1.063	0.738	1.045
PMF	0.993	1.823	0.893	1.415	1.100	1.850
NARRE	0.958	1.299	0.836	1.139	0.931	1.222
NRT	0.821	1.119	0.735	1.027	0.702	0.976
RippleNet	0.842	1.188	0.810	1.075	0.831	1.189
GCMC	0.981	1.166	0.950	1.117	0.673	1.013
LightGCN	1.153	1.513	1.093	1.452	1.057	1.365
KEGNN	**0.788**	**1.010**	**0.716**	**1.018**	**0.662**	**0.904**
Improvements	4.02%	9.74%	2.59%	0.88%	1.63%	7.38%

abstractive tips. The generated tips can been seen as explanations.

- *GCMC* [40]: is an GNN based recommendation method, which adopts GCN [17] encoder on the user-item interaction graph for rating matrix completion.
- *LightGCN* [41] is GCN-based recommendation method which only keeps neighborhood aggregation in GCN and achieved competitive performance.
- *RippleNet* [25]: is an end-to-end knowledge-aware recommendation framework, by propagating user preferences over knowledge entities. It combines both path-based and embedding-based methods for knowledge-aware recommendation.
- *KEGNN*: This is our proposed method for explainable recommendation.

The compared methods are divided into two groups. The first group of baselines can be evaluated with both the recommendation accuracy and the interpretability of generated explanation, including CTR, NARRE, NRT and our method KEGNN. The second group is only evaluated with the recommendation accuracy, including PMF, GCMC, LightGCN and RippleNet.

4.4 Implementation Details

All parameters are tuned with the validation set. After tuning process, the number of latent factors is set as: $k = 30$ for PMF, CTR, NARRE, and $k = 300$ for NRT's user latent factors, item latent factors and the word latent factors. For RippleNet[4], the embedding size is set as 16, and the number of hops is 2 . For GCMC, we employ the sum as accumulation function and symmetric normalization. The embedding size of LightGCN is 64. The drop ratio is tuned in {0.0, 0.1, $\cdots$, 0.8}.and the learning rate is 0.01. In our method, the embedding size of users, items and words is 64. The max number of node's neighbors is set as 30. The number of similar words given by ConceptNet is 10. The number of layers for the BiLSTM semantic encoder and the gru decoder is 2, the batch size for mini-batch training is 32. For optimization setting, we set the weight parameters $\lambda_r = \lambda_g = 1$. The regularization weight λ is 0.001. Each experiment is run three times, and the average results are reported in Section 5.

5 Experimental Results

5.1 Performance for Rating Prediction

The overall performance results of rating prediction on three datasets are shown in Table 3. The bold numbers indicate the best performance. The numbers with underlines indicate the second best one. The last line represents the improvements of our method over the second best one. From the results, we can see that our method KEGNN outperforms all compared methods in both MAE and RMSE for all datasets. In Electronic dataset, our method can achieve 9.74% improvements in RMSE and 4.02% improvements in MAE over the best competitor NRT. In Home&Kitchen daraset, the best competitor is NRT. We can have 2.59% improvements in MAE and 0.88% improvements in RMSE. In Music-Instrument dataset, the improvements can be about 7.38% in RMSE over NRT and 1.63% in MAE over GCMC. We performed paired t-test on the results between our method KEGNN and the best competitor respectively on the three datasets, and all results are significant with $p < 0.05$.

We observe that PMF has the worst performance in rating prediction. PMF only considers the rating information while all other methods utilize textual reviews for user preference leaning. This is verified by recent studies that joint modeling rating matrix and textual information are useful for improving recommendation performances. Although RippleNet leverages rational knowledge in knowledge graph, it does not show competitive performance. The reason might be that it takes more redundant entities as noises into representation learning, and requires informative (pre-defined) linked paths to propagate messages. However, it is not easy to extract and define paths for some domains.

5.2 Quality of Explanation Generation

We use ROUGE to evaluate the explanations in different granularities, similar to [20], [42]. Better ROUGE indicates higher similarities between the generated explanations and the ground truth user feelings (comments provided by users), which demonstrates better explainability. The evaluation results of explanation generation on the three datasets are respectively shown in Tables 4, 5, and 6. We report Recall, Precision, and F-measure of ROUGE-1, ROUGE-2, and ROUGE-L (in percentage). Our method KEGNN achives the best performance in Precision and F-measure for all datasets. Similar to the abstractive generation method NRT [20], we

[4] For datasets which have no linked relations in knowledge base/graph, we use item relations in meta data (e.g., brought_together, viewed_together, etc.) as the linked relations.

TABLE 4
ROUGE Evaluation on **Electronic** Dataset (%)

Method	ROUGE-1			ROUGE-2			ROUGE-L		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
CTR	14.93	11.14	12.76	2.36	1.96	2.14	12.74	10.45	11.48
NARRE	12.54	9.23	10.63	1.86	1.64	1.74	10.84	8.95	9.80
NRT	13.26	17.25	14.99	2.61	3.07	2.82	12.27	14.96	13.48
KEGNN	13.94	**17.82**	**15.64**	2.57	**3.27**	**2.88**	12.68	**15.83**	**14.08**

TABLE 5
ROUGE Evaluation on **Home&Kitchen** Dataset (%)

Method	ROUGE-1			ROUGE-2			ROUGE-L		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
CTR	9.45	15.83	11.84	1.25	1.91	1.27	8.95	14.91	11.18
NARRE	15.44	9.68	11.90	1.42	0.81	1.03	12.96	7.90	9.81
NRT	16.60	13.80	15.07	1.45	1.15	1.28	13.86	11.16	12.36
KEGNN	14.89	**17.56**	**16.12**	1.32	**1.93**	**1.57**	13.27	**16.54**	**14.73**

TABLE 6
ROUGE Evaluation on **Music-Instrument** Dataset (%)

Method	ROUGE-1			ROUGE-2			ROUGE-L		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
CTR	12.33	13.46	12.87	1.01	1.37	1.16	10.11	14.59	11.94
NARRE	13.52	11.25	12.28	1.42	1.09	1.23	15.36	11.48	13.14
NRT	11.75	14.42	12.95	1.35	1.97	1.60	12.64	15.22	13.81
KEGNN	11.36	**18.04**	**13.94**	1.31	**2.47**	**1.71**	12.71	**16.82**	**14.48**

do not have better performance than CTR or NARRE in Recall. It is because the compared methods CTR and NARRE are extraction-based methods, and they prefer to extract long sentences. However, NARRE have relatively poor performance in Precision as the extracted sentences might take many useless words and lead to unnecessarily verbose descriptions. The quality evaluation results indicate that our generated textual explanations are similar to ground truth comments associated with the rating behavior, and unfold the implicit users' intentions behind the rating behaviors. We also have case analysis and studies for further investigation of the explainability in Section 5.4.

5.3 Ablation Study

To inspect the influences of the important components of KEGNN, we conduct the ablation study by removing the knowledge enhanced representation learning (w/o Knowledge), user behavior learning and reasoning (w/o UBLR), the copy mechanism in the explanation generation module (w/o Copy), and the explanation generation module (w/o Gen). We design two variants for the hierarchical collaborative filtering component to verify the effectiveness of hierarchical neural layers (w/o hierarchical layers) and relation-aware layer (w/o relation layer). KEGNN w/o hierarchical layers only adopt one neural layer to perform the final rating prediction (only Eq. (6)). KEGNN w/o relation layer removes the relation representation in hierarchical collaborative filtering and the explanation generation module. In order to compare the influential effects of knowledge enhanced representation learning with pre-trained language models like BERT [43], we also implement one variant version with Bert, i.e., KEGNN with Bert, in which the knowledge-enhanced semantic representation learning is replaced with Bert semantic embedding.

The ablation results for rating prediction are shown in Table 7. We notice that the investigated components all contribute great improvements to our proposed method. First, we can see that the performance decreases sharply when removing the whole explanation generation module. We use the multi-task learning to perform rating prediction and explanation generation, and the two tasks have shared representation learning. During the multi-task learning, the model uses all of the available data across the different tasks to learn generalized representations of the data that are useful in multiple contexts. This observation validate that the explanation generation actually can help improve the effectiveness of rating prediction. Second, we find that KEGNN with Bert can not have good performance as KEGNN which integrated knowledge-enhanced representation learning. It indicates the influential effects of knowledge-enhanced representation learning, which can have better semantic representation learning with enhanced semantic concepts. In addition, we observe that KEGNN w/o Knowledge and KEGNN w/o UBLR have relatively inferior performance on the three datasets. It indicates the significant contributions of the knowledge-enhanced semantic representation learning module and

TABLE 7
Ablation Study for Rating Prediction

Method	Electrtonics		Home&Kitchen		Music-Instruments	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
KEGNN	0.788	1.010	0.716	1.018	0.662	0.904
KEGNN w/o Knowledge	0.797	1.039	0.768	1.155	0.688	0.915
KEGNN w/o UBLR	0.808	1.076	0.748	1.037	0.693	0.924
KEGNN w/o Copy	0.793	1.032	0.739	1.032	0.675	0.911
KEGNN w/o Gen	0.810	1.101	0.717	1.118	0.700	0.977
KEGNN w/o Hierarchical Layers	0.761	1.098	0.719	1.110	0.704	0.971
KEGNN w/o Relation-aware layer	0.857	1.284	0.740	1.121	0.694	1.037
KEGNN with Bert	0.855	1.115	0.818	1.041	0.765	1.013

TABLE 8
Ablation Study for ROUGE Evaluation on **Electronic** Dataset (%)

Method	ROUGE-1			ROUGE-2			ROUGE-L		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
KEGNN	13.94	17.82	15.64	2.57	3.27	2.88	12.68	15.83	14.08
KEGNN w/o Knowledge	12.27	16.63	14.12	2.14	2.98	2.49	10.84	14.34	12.35
KEGNN w/o UBLR	13.72	11.06	12.24	2.48	1.82	2.10	12.46	10.38	11.33
KEGNN w/o Copy	12.64	17.26	14.59	2.18	3.14	2.57	11.02	15.52	12.89
KEGNN w/o Hierarchical Layers	7.14	12.42	9.07	0.63	1.21	0.83	6.81	11.71	8.61
KEGNN w/o Relation-aware layer	6.01	16.87	8.86	0.41	1.34	0.63	5.93	16.57	8.73
KEGNN with Bert	7.06	17.19	10.00	0.82	2.34	1.22	6.74	15.36	9.37

TABLE 9
Ablation Study for ROUGE Evaluation on **Home&Kitchen** Dataset (%)

Method	ROUGE-1			ROUGE-2			ROUGE-L		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
KEGNN	14.89	17.56	16.12	1.32	1.93	1.57	13.27	16.54	14.73
KEGNN w/o Knowledge	14.62	13.29	13.92	1.24	1.23	1.23	13.10	11.63	12.32
KEGNN w/o UBLR	11.47	16.33	13.48	1.08	1.56	1.28	10.36	14.36	12.04
KEGNN w/o Copy	10.04	17.39	12.73	0.90	1.84	1.21	9.43	16.23	11.92
KEGNN w/o Hierarchical Layers	8.18	12.56	9.91	0.66	1.16	0.84	7.74	11.75	9.33
KEGNN w/o Relation-aware layer	7.30	14.77	9.77	0.57	1.43	0.82	6.96	13.95	9.29
KEGNN with Bert	5.92	16.63	8.73	0.63	1.77	0.93	5.70	15.98	8.40

the user behavior learning module (UBLR). As discussed before, the knowledge enhanced module leverages the common-sense semantic knowledge base to boost the semantic representation learning. The UBLR component exploits semantic enhanced knowledge to perform knowledge reasoning over user behavior graph and learn comprehensive representations of users, items and user-item interactions. Third, we can see that the performance of the two variants KEGNN w/o hierarchical layers and KEGNN w/o relation layer decreases on the three datasets which can validate the effectiveness of the hierarchical layers and the relation-aware layer.

In addition, we perform ablation studies on quality of explanation generation (explainability). The ablation results for explanation generation evaluation on the three datasets are summarized respectively in Tables 8, 9, and 10[5]. In general we can see that the ROUGE values decrease when removing the components. It indicates the designed components can contribute to explanation generation. We can see the ROUGE-Recall metrics significantly decrease when removing the copy mechanism (KEGNN w/o Copy). As we expected, the copy mechanism have important effects on explanation generation. KEGNN with Bert have poor performance in ROUGE-Recall. It might be because the pre-training models prefer to have general representations and generate general texts. For ROUGE-Precision metrics, we observe that the ROUGE-Precision performance of KEGNN on the Electronic dataset decreases sharply when removing the UBLR, which demonstrates the considerable effects of the UBLR component for user behavior modeling and accurate explanation generation. Although the user behavior graph from the Electronic dataset is large, and the UBLR module can have comprehensive understanding of user behaviors through information propagation and multi-hop reasoning over the large user behavior graph. KEGNN with Bert has relatively good performance in ROUGE-Precision. It is within our expectation as pre-training language models have had influential effects on similar semantic learning in recent studies. However, its generations are very general and similar as discussed in ROUGE-Recall

[5] KEGNN w/o Gen does not have generation results as it only keeps the rating prediction by removing the generation part.

TABLE 10
Ablation Study for ROUGE Evaluation on **Music-Instrument** Dataset (%)

Method	ROUGE-1			ROUGE-2			ROUGE-L		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
KEGNN	11.36	18.04	13.94	1.31	2.47	1.71	12.71	16.82	14.48
KEGNN w/o Knowledge	10.65	16.21	12.85	1.03	1.85	1.32	10.15	15.38	12.23
KEGNN w/o UBLR	11.21	17.82	13.76	1.23	2.21	1.58	12.24	16.48	14.05
KEGNN w/o Copy	10.79	15.85	12.84	1.12	1.79	1.38	10.22	14.95	12.14
KEGNN w/o Hierarchical Layers	8.97	17.88	11.95	0.82	1.79	1.13	8.18	16.08	10.85
KEGNN w/o Relation-aware layer	8.65	16.08	11.25	0.80	1.58	1.07	7.92	14.59	10.27
KEGNN with Bert	6.85	15.49	9.50	0.79	1.66	1.07	6.51	14.49	8.99

metrics. Therefore, its F metrics are low. In addition, we observe that KEGNN w/o hierarchical layers and KEGNN w/o relation layer have inferior ROUGE scores. It shows that the hierarchical layers can learn better representations which can have considerable effects on further explanation generation. And the relation representation performs important roles in high-quality explanation generation.

5.4 Case Study

We select some cases to investigate the explainability of the generated explanations. The selected cases are all from the test set. Therefore, the ground-truth texts are hidden during explanation generation. The exemplary cases on the three datasets are shown in Table 11. The ”Ground” line indicates the real reviews given by users. From the case studies, we can see that the explanations generated by our method indicate the users’ options and purchasing reasons for the rated item. The explainable concepts and aspects are highlighted in bold italic fonts, which indicates the potential intents of user behavior and demonstrates the interpretability of the recommendation results. For example, the ground truth for the first case in Table 11 is ”great case for a great price.” and the real rating is 5.0. The predicted rating is 4.71, and the generated explanation is ”great case with great price.” is very similar with the real textual review and convey the explainable concepts (great price) of the user behavior. For the negative opinions, the generated explanations also convey the negative sentiment. For instance, the last case in Home&Kitchen dataset is ”these were quite expensive without good quality” with 1.0 rating. The generated contexts conveys the negative sentiments (”disappointed”) and the reasons ”worried about the quality”. For negative cases, we can generate the correct sentiment explanations although the predicted rating number is not very similar. This might be due to user bias in rating. This observation might verify the importance of explainable recommendation which provides intuitive explanations with the simple rating number.

TABLE 11
Exemplary Cases of the Predicted Rating and the Generated Explanations on the Three Datasets. The Ground Line Indicates the Real Textual Review Given by Users. The Bold Italic Words Indicate the Explainable Aspects Behind the Rating Behavior.

Dataset	Rating	Generated Explanations
Electronics	5.0	Ground: great case for a great price.
	4.71	Our: great case with ***great price***.
	3.0	Ground: good price but less than stable.
	4.36	Our: this cable ***isnt a good quality***.
	1.0	Ground: what can you expect from a cheap hdmi cable?
	3.17	Our: This cable ***does not work***.
Home&Kitchen	5.0	Ground: 2nd waffle maker; this is the best.
	4.82	Our: I have owned this waffle maker for years and it is a ***great***.
	5.0	Ground: 15 years and still going strong.
	4.73	Our: I have been using these ***for years*** now and am overjoyed with them.
	1.0	Ground: these were quite expensive without good quality.
	2.67	Our: a bit ***disappointed*** with these and worried about the ***quality***.
Music-Instrument	5.0	Ground: Very solid keyboard stand for the price!
	4.81	Our: This is a very ***versatile*** stand with ***good price***.
	4.0	Ground: nice strap for the money.
	4.48	Our: great for the **price**.
	1.0	Ground: too small for my finger
	3.68	Our: the product is ***smaller*** than i needed.

In addition, we conduct another case study to investigate the effects of knowledge base on the explanations. Table 12 illustrates the example case about the relationship between the enhanced word-level concepts retrieved from knowledge base and the generated explanations. This case result is to investigate whether the enhanced KB words will have effects (relevant or occurring) on the generated explanations. We select one test case (the user ID is 1178 and the test item ID is 698) from the Music-Instrument dataset. The real rating r_{ij} is 5.0 and the predicted rating $\hat{r}_{ij}$ is 4.7. For the history rated items $I^{+}_{u=1178}$ of the user (1178), we compute

TABLE 12
Case Study of the Relationship Between the Enhanced Word-Level Concepts in Knowledge Base and the Generated Explanations

Case: ($u_i = 1178$, v_j = 698), $r_{ij} = 5.0$, $\hat{r}_{ij} = 4.7$		
$v_k \in I^{+}_{u=1178}$	Sim(C_{v_k}, C_{v_j})	Relevant Overlap KB Words
190	0.063	ukulelist, ukelele, banjouke, **size**, **good** well, perfect, **ukulele**, ukuleles, many
343	0.104	*buy*, large, *best*, **good**, price take, **value**, extremely, want, fashion
829	0.129	**ukulele**, *strings*, youtubian, get, ukelele tuned, tuning, aquila, sound, youtubes
Ground: The absolute best strings money can buy for a uke.		
Explanation: This is a very size for my **ukulele**, it is a **good value**.		

TABLE 13
RMSE Error Analysis at Different Rating for **Electronic** Dataset

Rating	5.0	4.0	3.0	2.0	1.0
Percentage	67.15%	16.60%	7.80%	4.00%	4.45%
NRT	0.769	0.659	1.372	2.312	3.079
KEGNN	0.683	**0.457**	1.196	2.143	2.910

Jaccard similarity of the enhanced word-level knowledge (concept word from ConceptNet) $C_{v_k}$and the enhanced word-level knowledge of the test item C_{v_j}. We select top-3 items (respectively 829, 343, 190) ranked at similarity scores of word-level knowledge. 10 relevant overlap KB words[6] are listed in the last column. We can see that the some KB words (in bold fonts) occur, or have similar semantic words in explanation (in bold fonts), e.g., size, ukulele, good, value. And some KB words also occur in the ground context (highlighted with italic fonts), e.g., *buy, best, strings*.

5.5 Error Analysis

In order to better understand the performance of our proposed method and study the phenomenon that the predicted rating for negative cases are not good (observed from 5.4), we also conduct a careful analysis to investigate the rating prediction error RSME over different types of real rating. In real datasets, the rating are given with the five numbers including $\{1.0, 2.0, 3.0, 4.0, 5.0\}$. Therefore, we calculate the rating performance RMSE of test datata for each rating number, and investigate our model performances for different ratings. The RMSE error analysis results at different rating numbers are shown respectively in Tables 13, 14, and 15 for Electronic dataset, Home&Kitchen dataset and Music-Instrument dataset.

We also include the analysis results for the best competitor NRT (Section 5.1) for reference. The ”Percentage” row indicates the percentage of test data each rating type accounts for. For all datasets, the largest proportion is at the rating number 5, with around 67.15% for Electronic dataset, 67.97% for Home&Kitchen dataset and 66.57% for Music-Instrument dataset. The test data at rating 4 take around 16.6% (Electronic), 16.35% (Home&Kitchen) and 18.16% (Music-Instrument) at the second place. The other three rating types (3.0, 2.0, 1.0) only takes a few percentages. We obverse that the RMSE for rating 5 and rating 4 are relatively small. The best result is obtained at rating 4. The phenomenon might be due to the imbalance in data proportion (as indicated the ”Percentage” row), and bias in user personalized opinions. This analysis observations might suggest that it is better to further focus on the few-shot rating types in future work.

6 Related Work

Traditional recommendation methods mainly contain collaborative filtering (CF) methods and content-based recommendation methods [1]. The underlying intuition idea behind collaborative filtering is to recommend similar items that a user might like on the basis of reactions by similar users. The content-based recommendation methods leverage the item features/attributes for recommendation. In recent decades, some model-based recommendation approaches have been proposed, e.g., matrix factorization [9]. Although model-based recommendation methods improve the recommendation accuracy, they hider the interpretability and transparency of the recommender system. Therefore, the new direction of explainable recommendation has been proposed [17], and several models have been designed to generate explainable recommendations while keeping a high recommendation accuracy [17], [20], [33].

TABLE 14
RMSE Error Analysis at Different Rating
for **Home&Kitchen** Dataset

Rating	5.0	4.0	3.0	2.0	1.0
Percentage	67.97%	16.35%	7.84%	4.32%	3.52%
NRT	0.715	0.607	1.343	2.139	2.959
KEGNN	0.660	**0.604**	1.397	2.181	3.029

Earlier explainable recommendations extracted features/aspects from user textual reviews and generated recommendation results with feature-level explanations [17], [18], [44], [45], [46]. Zhang *et al.* [17] proposed an explicit factor model (EFM) for explainable recommendation. They extracted explicit product features from user reviews and integrate user-feature and item-feature relations into a unified matrix factorization for recommendation. The feature-level explanations have been generated by mapping specific features with latent factors. Chen *et al.* [44] designed a tensor-matrix factorization method to learn user preferences over features, and proposed a hybrid ranking framework for recommendation with feature-level explanations. Wang *et al.* [45] proposed a multi-task learning method for explainable recommendation, in which the recommendation task and the explanation task are integrated via a joint tensor factorization.

Later, with the development of text generation in natural language processing [47], many methods generate human readable textual explanations, rather than feature-level explanations. For example, Seo *et al.* [48] designed interpretable convolutional neural networks for user preference modeling, and leveraged dual local and global attention to select and highlight informative words as explanations. Zhang *et al.* [19] proposed a neural attentional regression model for rating prediction, and introduced the attention mechanism to select the useful reviews as review-level explanations. Li *et al.* [20] proposed a deep learning framework named NRT which can simultaneously perform rating prediction and abstractive tips generation. Based on sequence-to-sequence (seq2seq) framework, gated neural networks [49] are adopted in abstractive tips generation to

TABLE 15
RMSE Error Analysis at Different Rating
for **Music-Instrument** Dataset

Rating	5.0	4.0	3.0	2.0	1.0
Percentage	66.57%	18.16%	9.22%	3.75%	2.31%
NRT	0.650	0.566	1.391	2.272	3.235
KEGNN	0.506	**0.486**	1.319	2.260	3.310

[6] Repeated words are deleted and only display 10 relevant words.

translate user and item latent representation into human readable contexts. Chen *et al.* [42] proposed a co-attentive multi-task learning model for explainable recommendation, in which an encoder-selector-decoder architecture has been introduced to model knowledge transfer between the recommendation task and explainable task.

Recently, knowledge graphs have been introduced into recommender system and employed to enhance the interpretability of recommendation methods with the propagation paths [25], [26], [50], [51], [52], [53], [54], [55]. For example, Wang *et al.* [25] incorporated knowledge graph to learn user preferences through preference propagation over knowledge entities linked in knowledge graph. The propagation paths demonstrated the reasoning process and improved the model transparency. Wang *et al.* [50] designed a knowledge-aware path recurrent network (KPRN) for recommendation, in which semantics of entities and relations have been combined to learn path representation. KPRN exploited the path sequential dependencies to infer user-item interactions, and endowed the model with a certain level of explainability by differentiating different paths. In addition, they further proposed the knowledge graph-based intent network (KGIN) [52], by exploring user intents behind a user-item interaction through attentive combination of item relevant relations within knowledge graph. [53] proposed relational collaborative filtering (RCF) to exploit multiple item relations in recommender systems, in which a two-level hierarchy of relation type and relation value is defined. Multiple relations between items have been exploited for user preference learning. This group of methods employ reasoning paths and relation links to demonstrate the model explanability. However, they have the strong prerequisites that the paths should be pre-defined and different linked relations exist in knowledge graph [54], [56]. The requirements cannot be flexible for all domains. For example, the data domains explored in the above studies are mostly from the movie and book domain in which multiple linked relation types exist in knowledge graph, and the meta-paths are easily pre-defined. However, they are not flexible to be applied for other datasets which have fewer links in knowledge graph, and meta-paths require human efforts. Our proposed method leverage the common-sense knowledge base ConceptNet, it is flexible to include external semantic knowledge through user history semantic comments (revealing real feelings and intents) for semantic knowledge enhanced representation learning, and mimic user rating behaviors to generate the potential semantic texts as explanations which can illustrate user potential reasons and improve the interpretability of the recommendation results.

7 Conclusion

In this manuscript, we propose knowledge enhanced graph neural networks (KEGNN) for explainable recommendation, in which semantic knowledge from the external knowledge base are exploited to enhance the representation learning of three sides including users, items and user-item interactions. We construct one user behavior graph and design a graph neural network based user behavior learning and reasoning module for comprehensive understanding of user behaviors. Finally, hierarchical neural layers are developed for precise rating prediction with human-like semantic explanations generated from the combination of GRU generator and the copy mechanism.

Although KEGNN has superior performance over the state-of-the-art methods, KEGNN cannot have high recommendation accuracy for negative cases due to few-shot issues as illustrated in the error analysis part (Section 5.5). In addition, KEGNN only exploits one type of relation in the user behavior graph, e.g., user-item interaction relation. In the future, we plan to leverage few-shot learning to improve the recommendation performance for few-shot rating types. And we will consider more relations within user behaviors when constructing the user behavior graph, and leverage multiple relations for user behavior learning.

References

[1] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[2] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, Berlin, Germany: Springer, vol. 1–35, 2010, pp. 1–35.

[3] 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, May 2014.

[4] H. Li, Y. Wang, Z. Lyu, and J. Shi, “Multi-task learning for recommendation over heterogeneous information network,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 1–1, Feb. 2020.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.

[6] Z. Lu, H. Wang, N. Mamoulis, W. Tu, and D. Cheung, “Personalized location recommendation by aggregating multiple recommenders in diversity,” *GeoInformatica*, vol. 21, pp. 1–26, 2017.

[7] Z. Lu, H. Li, N. Mamoulis, and D. W. Cheung, “HBGG: a hierarchical Bayesian geographical model for group recommendation,” in *Proc. SIAM Int. Conf. Data Mining*, 2017, pp. 372–380.

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

[9] R. Salakhutdinov and A. Mnih, “Probabilistic matrix factorization,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.

[10] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, 2019.

[11] H.-T. Cheng *et al.*, “Wide & deep learning for recommender systems,” in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[12] A. B. Arrieta *et al.*, “Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Inf. Fusion*, vol. 58, pp. 82–115, 2020.

[13] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, “Definitions, methods, and applications in interpretable machine learning,” *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 44, pp. 22071–22080, Oct. 2019.

[14] Y. Zhang and X. Chen, “Explainable recommendation: A survey and new perspectives,” *Found. Trends Inf. Retrieval*, vol. 14, no. 1, pp. 1–101, 2020.

[15] B. Knijnenburg, M. Willemsen, Z. Gantner, H. Soncu, and C. Newell, “Explaining the user experience of recommender systems,” *User Model. User-Adapted Interac.*, vol. 22, pp. 441–504, 2012.

[16] H. Cramer *et al.*, “The effects of transparency on trust and acceptance in interaction with a content-based art recommender,” *User Model. User-Adapt. Interact.*, vol. 18, pp. 455–496, 2008.

[17] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, “Explicit factor models for explainable recommendation based on phrase-level sentiment analysis,” in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 83–92.

[18] X. He, T. Chen, M.-Y. Kan, and X. Chen, “TriRank: Review-aware explainable recommendation by modeling aspects,” in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 1661–1670.

[19] C. Chen, M. Zhang, Y. Liu, and S. Ma, “Neural attentional rating regression with review-level explanations,” in *Proc. World Wide Web Conf.*, 2018, pp. 1583–1592.

[20] P. Li, Z. Wang, Z. Ren, L. Bing, and W. Lam, "Neural rating regression with abstractive tips generation for recommendation," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 345–354.

[21] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[22] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *NIPS 2014 Workshop Deep Learn.*, vol. abs/1412.3555, 2014. [Online]. Available: https://arxiv.org/abs/1412.3555

[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.*, 2019, pp. 2000–2010.

[24] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proc. World Wide Web Conf.*, 2019, pp. 3307–3313.

[25] H. Wang *et al.*, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 417–426.

[26] 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.*, 2019, pp. 151–161.

[27] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2019, pp. 5329–5336.

[28] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An open multilingual graph of general knowledge," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4444–4451.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Workshop Int. Conf. Learn. Representations*, 2013.

[30] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2005, pp. 2047–2052.

[31] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[33] 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. Discov. Data Mining*, 2019, pp. 950–958.

[34] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "Deepinf: Social influence prediction with deep learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2110–2119.

[35] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics* 2016, pp. 1631–1640.

[36] S. Ruder, "An overview of multi-task learning in deep neural networks," *Comput. Res. Repository*, vol. abs/1706.05098, 2017.

[37] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization Branches Out Post-Conf. Workshop ACL*, 2004, pp. 74–81.

[38] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn. 32 Papers*, 2016, pp. 280–290.

[39] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2011, pp. 448–456.

[40] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*.

[41] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 639–648.

[42] Z. Chen *et al.*, "Co-attentive multi-task learning for explainable recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 2137–2143.

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *North Amer. Chapter Assoc. Comput. Linguistics*, pp. 4171–4186, Jun. 2019.

[44] X. Chen, Z. Qin, Y. Zhang, and T. Xu, "Learning to rank features for recommendation over multiple categories," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2016, pp. 305–314.

[45] N. Wang, H. Wang, Y. Jia, and Y. Yin, "Explainable recommendation via multi-task learning in opinionated text data," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 165–174.

[46] Z. Ren, S. Liang, P. Li, S. Wang, and M. de Rijke, "Social collaborative viewpoint regression with explainable recommendations," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 485–494.

[47] A. Gatt and E. Krahmer, "Survey of the state of the art in natural language generation: Core tasks, applications and evaluation," *J. Artif. Int. Res.*, vol. 61, no. 1, pp. 65–170, Jan. 2018.

[48] S. Seo, J. Huang, H. Yang, and Y. Liu, "Interpretable convolutional neural networks with dual local and global attention for review rating prediction," in *Proc. 11th ACM Conf. Recommender Syst.*, 2017, pp. 297–305.

[49] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *EMNLP*, pp. 1724–1734, 2014.

[50] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2019, pp. 5329–5336.

[51] W. Ma *et al.*, "Jointly learning explainable rules for recommendation with knowledge graph," in *Proc. World Wide Web Conf.*, 2019, pp. 1210–1221.

[52] X. Wang *et al.*, "Learning intents behind interactions with knowledge graph for recommendation," in *Proc. World Wide Web Conf.*, 2021, pp. 878–887.

[53] 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. Retrieval*, 2019, pp. 125–134.

[54] K. Zhao *et al.*, "Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 239–248.

[55] H. Chen, Y. Li, X. Sun, G. Xu, and H. Yin, "Temporal meta-path guided explainable recommendation," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, 2021, pp. 1056–1064.

[56] Z. Chen *et al.*, "Towards explainable conversational recommendation," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2020, pp. 2994–3000.

Ziyu Lyu received the PhD degree in computer science from the University of Hong Kong, Hong Kong, in 2016. She is currently an associate professor with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. Her research interests include recommender systems, natural language processing, and machine learning.

Yue Wu received the BE degree from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2019. She is currently working toward the master degree at the Chongqing University of Posts and Telecommunications, Chongqing, China. Her research interests include natural language processing and computer vision.

Junjie Lai received the BE degree from Sichuan University, Chengdu, China, in 2017. He is currently working toward the master degree at the University of Science and Technology of China, Hefei, China. His research interests include natural language processing and recommender systems.

Min Yang received the BS degree from Sichuan University, Chengdu, China, in 2012, and the PhD degree from the University of Hong Kong, Hong Kong, in February 2017. She is currently an associate professor with the Shenzhen Institute of Advanced Technology, Chinese Academy of Science. Her current research interests include machine learning, deep learning, and natural language processing.

Chengming Li received the PhD degree from the Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan, in 2015. He is currently an associate professor with the School of Intelligent Systems Engineering, Sun Yat-sen University. His research interests include data mining, natural language processing, and Intelligent Internet.

Wei Zhou received the PhD degree from Chongqing University, Chongqing, China, in December 2015. He is currently an associate professor with the School of Big data and Software Engineering, Chonging University. His current research interests include recommender systems, machine learning, deep learning, and natural language processing.